



KERJA PRAKTIK – IF184801

Pembuatan Aplikasi ITS JobX: Platform Pengembangan Sumber Daya Mahasiswa dengan Metode On Job Training untuk Mendukung Kebijakan Kampus Merdeka

Pusat Pengembangan Karir dan Kewirausahaan Mahasiswa Institut Teknologi Sepuluh Nopember

Gedung SAC Kampus ITS JL. Arif Rahman Hakim Sukolilo Surabaya

Periode: 1 Mei 2020 - 31 Oktober 2020

Oleh:

Iqbaal Pratama Putra 05111840000021

Pembimbing Jurusan

Wahyu Suadi, S.Kom., M.Kom

Pembimbing Lapangan

Muchammad Nurif, SE., MT.

DEPARTEMEN TEKNIK INFORMATIKA

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya 2021

[Halaman ini sengaja dikosongkan]



Pembuatan Aplikasi ITS JobX: Platform Pengembangan Sumber Daya Mahasiswa dengan Metode On Job Training untuk Mendukung Kebijakan Kampus Merdeka

Pusat Pengembangan Karir dan Kewirausahaan Mahasiswa Institut Teknologi Sepuluh Nopember

Gedung SAC Kampus ITS JL. Arif Rahman Hakim Sukolilo Surabaya

Periode: 1 Mei 2020 - 31 Oktober 2020

Oleh:

Iqbaal Pratama Putra 05111840000021

Pembimbing Jurusan

Wahyu Suadi, S.Kom., M.Kom

Pembimbing Lapangan

Muchammad Nurif, SE., MT.

DEPARTEMEN TEKNIK INFORMATIKA

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya 2021

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN
KERJA PRAKTIK

**Pembuatan Aplikasi ITS JobX: Platform Pengembangan
Sumber Daya Mahasiswa dengan Metode On Job
Training untuk Mendukung Kebijakan Kampus Merdeka**

Oleh:

Iqbaal Pratama Putra

05111840000021

Mengetahui,
Pembimbing Kerja Praktik



Muchammad Nurif, SE., MT
NIP 19690614 199802 1 001

Menyetujui,
Dosen Pembimbing,
Kerja Praktik



Wahyu Suadi, S.Kom., M.Kom
NIP. 1971 10302002121001

SURABAYA, JUNI 2021

[Halaman ini sengaja dikosongkan]

**Pembuatan Aplikasi ITS JobX: Platform Pengembangan
Sumber Daya Mahasiswa dengan Metode On Job Training
untuk Mendukung Kebijakan Kampus Merdeka**

Nama Mahasiswa	: Iqbaal Pratama Putra
NRP	: 05111840000021
Departemen	: Informatika FTEIC-ITS
Pembimbing Jurusan	: Wahyu Suadi, S.Kom., M.Kom.
Pembimbing Lapangan	: Muchammad Nurif, SE., MT.

ABSTRAK

Diresmikannya program Kampus Merdeka oleh Menteri Pendidikan dan Kebudayaan sangat menguntungkan bagi mahasiswa dimana mahasiswa diberikan keleluasan untuk mengeksplor *passion* yang dimilikinya dengan memiliki hak belajar 3 semester di luar program studi yang sedang dijalani.

Institut Teknologi Sepuluh Nopember Surabaya belum memiliki platform yang cukup tepat untuk memaksimalkan program Kampus Merdeka, untuk itu, Pusat Pengembangan Karir dan Kewirausahaan Mahasiswa Institut Teknologi Sepuluh Nopember membutuhkan bekerja sama dengan Tim Developer ITS JobX membuat sebuah platform yang mana bisa memfasilitasi mahasiswa ITS dalam mengikuti program Kampus Merdeka, selain itu platform ini juga bisa digunakan untuk Perusahaan Luar yang ingin merekrut mahasiswa ITS sebagai mahasiswa magang ataupun pegawai tetap .

Aplikasi ini dibuat dengan menggunakan bahasa pemrograman web seperti PHP, CSS, HTML, dan Javascript dengan menggunakan DBMS MySQL Server. Aplikasi ini diharapkan dapat mempermudah pemantauan terhadap mahasiswa ITS mengenai keikutsertaan dalam program magang serta program ITS Youth Technopreneur.

Kata kunci: ITS JobX, Kampus Merdeka, Mahasiswa, Website

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah SWT karena berkat rahmat-Nya penulis dapat melaksanakan salah satu kewajiban sebagai mahasiswa Departemen Informatika Institut Teknologi Sepuluh Nopember Surabaya, yakni Kerja Praktik (KP) yang berjudul Pembuatan Aplikasi ITS JobX: Platform Pengembangan Sumber Daya Mahasiswa dengan Metode On Job Training untuk Mendukung Kebijakan Kampus Merdeka.

Kami menyadari masih ada kekurangan baik dalam pelaksanaan kerja praktik maupun penyusunan buku laporan ini. Namun, kami berharap buku laporan ini dapat menambah wawasan pembaca dan dapat menjadi sumber referensi. Kami mengharapkan kritik dan saran yang membangun untuk kesempurnaan buku laporan kerja praktik ini.

Dalam penyusunan laporan ini tidak lepas dari bimbingan dan dukungan beberapa pihak, untuk itu kami menyampaikan terima kasih pada:

1. Bapak Wahyu Suadi, S.Kom., M.Kom. selaku dosen pembimbing kerja praktik.
2. Bapak Ary Mazharuddin Shiddiqi, S.Kom., M.Comp., Ph.D selaku koordinator Kerja Praktik.
3. Bapak Arief Abdurrahman dan Muchammad Nurif, S.E., M.T. selaku dosen pembimbing eksternal yang telah

memberikan kami kesempatan untuk belajar di Subdirektorat PK2M.

4. Dea Adivanessa Aura, Rafi Nizar Abiyyi dan Abdur Rohman selaku anggota Tim ITS JobX.
5. Serta keluarga dan teman-teman yang turut memberi dukungan sehingga proses pengerjaan terselesaikan dengan baik

Surabaya, Juni 2021

Iqbaal Pratama Putra

DAFTAR ISI

LEMBAR PENGESAHAN	5
ABSTRAK	8
KATA PENGANTAR	9
DAFTAR ISI	11
DAFTAR TABEL	14
DAFTAR GAMBAR	15
DAFTAR KODE	17
BAB I PENDAHULUAN	18
1.1. Latar Belakang	18
1.2. Tujuan	19
1.3. Manfaat	19
1.4. Rumusan Masalah	20
1.5. Lokasi dan Waktu Kerja Praktik	20
1.6. Metodologi Kerja Praktik	20
1.6.1. Perumusan Masalah	21
1.6.2. Studi Literatur	21
1.6.3. Analisis dan Perancangan Sistem	22
1.6.4. Implementasi Sistem	22
1.6.5. Pengujian dan Evaluasi	22
1.6.6. Kesimpulan dan Saran	23
1.7. Sistematika Laporan	23
1.7.1. Bab I Pendahuluan	23

1.7.2. Bab II Profil Perusahaan	23
1.7.3. Bab III Tinjauan Pustaka	24
1.7.4. Bab IV Desain Sistem	24
1.7.5. Bab V Implementasi Sistem	24
1.7.6. Bab VI Pengujian dan Evaluasi	25
1.7.7. Bab VII Kesimpulan dan Saran	25
BAB II PROFIL PERUSAHAAN	27
2.1. Profil Pusat Pengembangan Karir dan Kewirausahaan Mahasiswa Institut Teknologi Sepuluh Nopember Surabaya	27
2.2. Visi Misi Pusat Pengembangan Karir dan Kewirausahaan Mahasiswa Institut Teknologi Sepuluh Nopember Surabaya	28
2.2.1. Visi	28
2.2.2. Misi	29
BAB III TINJAUAN PUSTAKA	31
3.1. Laravel	31
3.2. Visual Studio Code	31
3.3. MySQL	32
3.4. PHP	32
3.5. HTML	32
3.6. CSS	33
3.7. Javascript	33
3.8. Git	34

3.9. Midtrans	34
BAB IV DESAIN SISTEM	36
4.1. Desain Data Model	36
4.2. Desain Data Flow Diagram	50
BAB V IMPLEMENTASI SISTEM	52
5.1. Implementasi Source Code	52
5.2. Implementasi Antarmuka Pengguna	136
BAB VI PENGUJIAN DAN EVALUASI	145
6.1. Tujuan Pengujian	145
6.2. Kriteria Pengujian	145
6.3. Skenario Pengujian	145
6.4. Evaluasi Pengujian	148
BAB VII KESIMPULAN DAN SARAN	152
7.1. Kesimpulan	152
7.2. Saran	152
DAFTAR PUSTAKA	154
BIODATA PENULIS	156

DAFTAR TABEL

Tabel 6.4.1 Hasil Pengujian

150

DAFTAR GAMBAR

Gambar 4.1.1 Desain Data Model	36
Gambar 4.1.2. Model tabel admins	37
Gambar 4.1.3. Model tabel students	38
Gambar 4.1.4. Model tabel guests	38
Gambar 4.1.5. Model tabel employers	39
Gambar 4.1.6. Model tabel mentors	39
Gambar 4.1.7. Model tabel jobs	40
Gambar 4.1.8. Model tabel seminars	41
Gambar 4.1.9. Model tabel services	41
Gambar 4.1.10. Model tabel job_categories	42
Gambar 4.1.11. Model tabel seminar_categories	42
Gambar 4.1.12. Model tabel job_student	43
Gambar 4.1.13. Model tabel guest_services	43
Gambar 4.1.14. Model tabel investee	44
Gambar 4.1.15. Model tabel investasi_project	44
Gambar 4.1.16. Model tabel investasi_funding	45
Gambar 4.1.17. Model tabel investasi	46
Gambar 4.1.18. Model tabel investasi_student	47
Gambar 4.1.19. Model tabel progres_project	47
Gambar 4.1.20. Model tabel i_y_t_batches	48
Gambar 4.1.21. Model tabel investasi_iyt	48
Gambar 4.1.22. Model tabel iyt_mentorings	49
Gambar 4.1.23. Model tabel notulensi	49
Gambar 4.1.24. Model tabel order	50

Gambar 4.2.1. Data Flow Diagram Level 0	50
Gambar 5.2.1. Halaman Pendaftaran User Employer	135
Gambar 5.2.2. Halaman Pendaftaran User Employer	135
Gambar 5.2.3. Halaman Verifikasi Pekerjaan	136
Gambar 5.2.4. Halaman Pekerjaan yang ditawarkan	136
Gambar 5.2.5. Halaman Melamar Pekerjaan bagi Mahasiswa	137
Gambar 5.2.6. Halaman Cek Pekerjaan yang disetujui ditampilkan	137
Gambar 5.2.7. Halaman Cek Lamaran Pekerjaan	138
Gambar 5.2.8. Halaman Pengajuan Pelaksanaan Seminar	138
Gambar 5.2.9. Halaman Cek Seminar yang tersedia	139
Gambar 5.2.10. Halaman Pembuatan Jasa	139
Gambar 5.2.11. Halaman Jasa Tersedia	140
Gambar 5.2.12. Halaman Detail Jasa	140
Gambar 5.2.13. Halaman Persetujuan Jasa	141
Gambar 5.2.14. Halaman Permintaan Jasa Masuk	141
Gambar 5.2.15. Halaman Permintaan Jasa yang sedang berjalan	142
Gambar 5.2.16. Halaman ITS Youth Technopreneur	142

DAFTAR KODE

Kode 5.1.1 Kode Admin Controller	53
Kode 5.1.2 Kode Dashboard Controller	68
Kode 5.1.3 Kode Dashboard Employer Controller	81
Kode 5.1.4 Kode Dashboard Guest Controller	85
Kode 5.1.5 Kode Dashboard Investee Controller	94
Kode 5.1.6 Kode IYT Controller	100
Kode 5.1.7 Kode Dashboard Student Controller	104
Kode 5.1.8 Kode Employer Controller	109
Kode 5.1.9 Kode Guest Controller	113
Kode 5.1.10 Kode Investasi Controller	122
Kode 5.1.11 Kode Job Controller	125
Kode 5.1.12 Kode Seminar Controller	127
Kode 5.1.13 Kode Services Controller	129
Kode 5.1.14 Kode Student Controller	134

BAB I

PENDAHULUAN

1.1. Latar Belakang

Pengembangan karir dan kewirausahaan Mahasiswa merupakan hal yang penting bagi setiap Perguruan Tinggi, untuk itu perlu adanya sebuah platform yang membantu Mahasiswa dalam menemukan minat karir dan kewirausahaannya. ITS sendiri sebenarnya telah memiliki platform sedemikian rupa, namun platform tersebut hanya dapat diakses oleh alumni ITS dan orang luar ITS yang telah berlangganan. Padahal untuk bisa melatih diri dalam lingkungan kerja, melatih jiwa entrepreneurship perlu dibangun sejak menjadi mahasiswa

Pentingnya pengembangan karir dan kewirausahaan untuk mahasiswa juga sedang disoroti Menteri Pendidikan dan Kebudayaan Republik Indonesia, Nadiem Makarim. Yang mana pada Januari 2020 meluncurkan kebijakan baru yang bernama Kampus Merdeka. Kebijakan ini memang dikhususkan untuk mahasiswa Perguruan Tinggi dimana salah satu poin utamanya yaitu mahasiswa diberikan hak belajar selama 3 semester diluar program studi yang diambil. Kebijakan ini sedang dalam proses penyesuaian untuk diterapkan di seluruh Perguruan Tinggi Indonesia, salah satunya ITS.

Berlatar belakang permasalahan tersebut, perlu adanya sebuah platform milik ITS yang memuat informasi magang, lowongan kerja part-time, lowongan kerja proyek ataupun freelance dan program pengembangan kewirausahaan agar Mahasiswa ITS lebih baik dalam mempersiapkan bekal kehidupan pasca kampus, ITS JobX hadir sebagai sarana Mahasiswa ITS dalam mengeksplorasi berbagai jenis pekerjaan dari berbagai perusahaan sesuai bidang masing-masing dan pembinaan start-up mahasiswa.

1.2.Tujuan

Tujuan kerja praktik ini adalah untuk menyelesaikan kewajiban kuliah kerja praktik di Institut Teknologi Sepuluh Nopember dengan beban dua SKS. Selain itu juga untuk membantu Direktorat Kemahasiswaan khususnya P2K2M untuk mengelola program pengembangan karir dan kewirausahaan mahasiswa.

1.3.Manfaat

Manfaat dari kerja praktik ini sama dengan tujuan diatas yaitu untuk membantu Direktorat Kemahasiswaan yaitu P2K2M mengelola program pengembangan karir dan kewirausahaan mahasiswa terutama membangun platform Pengembangan Sumber Daya Mahasiswa dengan Metode On Job Training untuk Mendukung Kebijakan Kampus Merdeka.

1.4. Rumusan Masalah

Berikut ini rumusan masalah pada kerja praktik pembuatan Aplikasi ITS JobX:

1. Bagaimana implementasi Aplikasi ITS JobX?
2. Bagaimana alur Penggunaan Aplikasi ITS JobX?

1.5. Lokasi dan Waktu Kerja Praktik

Kerja praktik ini dilaksanakan pada waktu dan tempat sebagai berikut:

Lokasi : Online

Waktu : 1 Mei 2020 – 31 Oktober 2020

Hari Kerja : Senin - Minggu

Jam Kerja : 08.00 WIB – 17.00 WIB

Kerja praktik dilaksanakan secara daring dikarenakan pandemi COVID-19

1.6. Metodologi Kerja Praktik

Tahapan pengerjaan kerja praktik dapat dijabarkan sebagai berikut:

1.6.1. Perumusan Masalah

Dalam tahap ini kami perlu mengetahui permasalahan apa saja yang terjadi dan dapat diselesaikan atau dioptimasi. Lalu kami juga perlu mengetahui semua kebutuhan dalam permasalahan tersebut.

Untuk mengetahui kebutuhan-kebutuhan fungsional dan non-fungsional dari aplikasi ini, kami mengikuti rapat-rapat bersama Tim ITS JobX serta perwakilan dari Pusat Pengembangan Karir dan Kewirausahaan Mahasiswa Institut Teknologi Sepuluh Nopember Surabaya. Pada rapat tersebut, dijelaskan kebutuhan-kebutuhan dari sistem yang akan dibuat. Setelah rapat tersebut, berdasarkan kebutuhan sistem yang sebelumnya ditemukan, kebutuhan tersebut diimplementasikan.

1.6.2. Studi Literatur

Setelah ditentukan kebutuhan sistem dari aplikasi ITS JobX ini, seperti database, bahasa pemrograman, dan serta tools tambahan yang akan digunakan, dilakukan studi literatur mengenai cara implementasinya, dalam hal ini tinjauan dilakukan

terhadap framework Laravel 7, software database MySQL dengan bahasa pemrograman PHP, HTML, CSS, dan Javascript. Pada tahap ini dilakukan proses pencarian, pembelajaran, pengumpulan dan pemahaman informasi serta literatur yang berkaitan untuk membantu dalam implementasi segala bentuk code atau query yang ada.

1.6.3. Analisis dan Perancangan Sistem

Setelah melakukan studi literatur, untuk merancang sistem yang baik perlu adanya sebuah desain arsitektur sistem. Pada website ini tim developer ITS JobX setuju untuk menggunakan arsitektur desain MVC (Model - View - Controller).

1.6.4. Implementasi Sistem

Implementasi merupakan realisasi dari tahap perancangan. Pada tahap ini kami membuat website untuk memenuhi kebutuhan sistem.

1.6.5. Pengujian dan Evaluasi

Pengujian dilakukan dengan menguji fitur-fitur yang telah dibuat. Website yang sudah dikembangkan akan diperiksa oleh pembimbing

lapangan kerja praktik apakah fitur-fitur yang telah dibuat di dalamnya sesuai dengan yang dibutuhkan termasuk runtime dari fungsi yang dibuat. Hal ini akan menentukan apakah hasil evaluasi terhadap sistem sudah sesuai dengan tujuan dan kebutuhan sistem yang sudah di analisis.

1.6.6. Kesimpulan dan Saran

Pada bab ini, dipaparkan kesimpulan yang dapat diambil dan juga saran dalam pengerjaan kerja praktik.

1.7.Sistematika Laporan

Laporan kerja praktik ini terdiri dari tujuh bab dengan rincian sebagai berikut:

1.7.1. Bab I Pendahuluan

Pada bab ini dijelaskan tentang latar belakang permasalahan, tujuan, waktu pelaksanaan, serta sistematika pengerjaan kerja praktik dan juga penulisan laporan kerja praktik.

1.7.2. Bab II Profil Perusahaan

Bab ini akan menjelaskan secara rinci tentang profil perusahaan Pusat Pengembangan Karir dan Kewirausahaan Mahasiswa Institut Teknologi Sepuluh Nopember Surabaya, tempat saya melaksanakan kerja praktik.

1.7.3. Bab III Tinjauan Pustaka

Pada bab ini, dijelaskan mengenai tinjauan Pustaka dan literatur yang digunakan dalam penyelesaian kerja praktik di Pusat Pengembangan Karir dan Kewirausahaan Mahasiswa Institut Teknologi Sepuluh Nopember Surabaya.

1.7.4. Bab IV Desain Sistem

Pada bab ini berisi uraian desain data model serta alur data yang digunakan dalam sistem.

1.7.5. Bab V Implementasi Sistem

Pada bab ini berisi uraian tahap - tahap yang dilakukan untuk proses implementasi sistem.

1.7.6. Bab VI Pengujian dan Evaluasi

Pada bab ini berisi hasil uji coba dan evaluasi dari aplikasi yang telah dikembangkan selama pelaksanaan kerja praktik

1.7.7. Bab VII Kesimpulan dan Saran

Pada bab ini, akan dipaparkan kesimpulan yang dapat diambil dan juga saran selama pengerjaan kerja praktik.

[Halaman ini sengaja dikosongkan]

BAB II

PROFIL PERUSAHAAN

2.1. Profil Pusat Pengembangan Karir dan Kewirausahaan Mahasiswa Institut Teknologi Sepuluh Nopember Surabaya

Sejak berdiri hingga saat ini PK2M ITS telah berperan cukup signifikan bagi ITS secara institusional maupun bagi mahasiswa dan alumni. Data hasil pelacakan alumni dan kerjasama industri dapat dimanfaatkan untuk pengukuran kepuasan pengguna dan digunakan untuk akreditasi institusi. Demikian juga dengan layanan psikologi sangat membantu untuk memetakan kondisi awal mahasiswa ITS sehingga dapat didesain proses pendidikan yang tepat selama mahasiswa tersebut berada di ITS. Selain itu bimbingan psikologis dan karir sangat membantu mahasiswa dalam merancang jenjang karir yang ingin dibangun setelah lulus dari ITS.

Berbagai informasi dunia kerja yang sesuai dengan kompetensi yang dipelajari sangat membantu menjembatani mahasiswa untuk berkarir dengan penuh perencanaan sejak awal. PK2M ITS juga memetakan jumlah mahasiswa ITS yang terbantu dengan keberadaan PK2M ITS dari aspek memperoleh kesempatan melakukan konsultasi psikologi dan bimbingan karir, memperoleh informasi peluang karir baik online maupun offline melalui bursa karir ITS. Kerjasama PK2M

ITS dengan Industri khususnya dalam bursa karir ITS juga mendapatkan respon positif.

Saat ini PK2M ITS terus berbenah untuk meningkatkan kualitas layanan guna mencapai visi misi yang telah ditetapkan. Salah satu hal yang sedang dilaksanakan adalah meningkatkan kualitas sistem informasi dan komunikasi untuk aplikasi kerjasama industri, info layanan, pelacakan lulusan hingga pelaksanaan bursa karir ITS. Program pengembangan yang juga dilakukan adalah peningkatan kualitas dan sistem kerja praktek (internship), kuliah tamu dan turunannya. Berbagai layanan psikologi juga terus ditingkatkan berdasarkan hasil survey kebutuhan baik untuk mahasiswa ataupun calon wisudawan. PK2M ITS saat ini bergabung dan bekerjasama dengan ICCN (Indonesia Career Center Network) sebuah organisasi yang mewadahi pusat karir yang ada di Perguruan Tinggi di Indonesia. Melalui kerjasama ini, PK2M ITS banyak belajar dan berbagi dengan berbagai pusat pengembangan karir di berbagai perguruan tinggi di Indonesia.

2.2. Visi Misi Profil Pusat Pengembangan Karir dan Kewirausahaan Mahasiswa Institut Teknologi Sepuluh Nopember Surabaya

2.2.1. Visi

Menjadi lembaga bereputasi internasional yang mampu meningkatkan kompetensi calon lulusan sesuai

tuntutan dunia kerja dan mengembangkan kemampuan mahasiswa dalam bidang kewirausahaan

2.2.2. Misi

1. Menyelenggarakan program peningkatan kompetensi calon lulusan sesuai tuntutan dunia kerja internasional
2. Mendukung program penyusunan dan penyelenggaraan pendidikan berdasarkan masukan dari alumni dan pengguna lulusan
3. Menyelenggarakan program layanan bagi mahasiswa untuk berkarir
4. Mengembangkan layanan khusus bagi dunia kerja dalam memperoleh informasi kompetensi setiap program studi dan menyelenggarakan rekrutmen calon pencari kerja sesuai kompetensi yang diperlukan
5. Menyelenggarakan program pelayanan dan pembinaan kewirausahaan mahasiswa untuk meningkatkan kompetensi

[Halaman ini sengaja dikosongkan]

BAB III

TINJAUAN PUSTAKA

Pada bab ini, akan dijelaskan mengenai dasar teori yang digunakan selama proses kerja praktik.

3.1. Laravel

Laravel merupakan framework PHP yang banyak digunakan oleh *Web Developer* karena kinerja, fitur serta skalabilitas yang dimiliki, framework ini mengikuti struktur MVC (Model-View-Controller) yang mana merupakan metode aplikasi yang memisahkan antara data dari tampilan berdasarkan komponen-komponen aplikasi. Selain itu, framework ini banyak disukai oleh para *Web Developer* juga karena juga menyediakan fitur bawaan seperti autentifikasi, mail, perutean, sesi dan daftar berjalan^[1].

3.2. Visual Studio Code

Visual Studio Code merupakan kode editor yang ringan, dapat digunakan untuk membuat aplikasi web. Aplikasi ini termasuk aplikasi lintas platform yang dapat digunakan di berbagai sistem operasi Windows, Linux, dan Mac OS, selain itu aplikasi ini dapat digunakan untuk berbagai macam Bahasa pemrograman seperti Java, Javascript, Go, C++ dan banyak yang lain^[2].

3.3. MySQL

MySQL merupakan salah satu *Database Management System* yang cukup terkenal karena sudah sering digunakan karena sifatnya yang *open source*. MySQL sendiri menggunakan perintah dasar SQL (Structured Query Language) dan MySQL ini sebagai penghubung antara perangkat lunak aplikasi dengan database server^[3].

3.4. PHP

PHP atau singkatan dari PHP: Hypertext Preprocessor adalah salah satu bahasa pemrograman populer yang biasa digunakan dalam pembuatan dan pengembangan sebuah website. PHP sendiri adalah bahasa yang sering digunakan untuk fungsi komunikasi pada sisi server. PHP sendiri menjadi salah satu bahasa pemrograman yang populer dikarenakan memiliki beberapa kelebihan antara lain mudah dipelajari, banyak digunakan, hemat biaya, ada banyak komunitasnya dan terintegrasi dengan database^[4].

3.5. HTML

HTML atau singkatan dari Hypertext Markup Language adalah bahasa computer yang terdiri dari sekumpulan kode dengan tujuan mengatur struktur dan menyajikan informasi yang mana HTML ini biasa digunakan untuk membuat struktur halaman dari website. HTML sendiri terdiri dari tiga anggota inti yaitu tag yang digunakan sebagai awalan instruksi yang dibaca web browser, element yang

merupakan keseluruhan kode yang terdiri dari tag pembuka (<>) hingga tag penutup (</>), dan yang terakhir attribute yang merupakan informasi atau perintah tambahan yang berada di dalam elemen^[5].

3.6. CSS

CSS atau singkatan dari Cascading Style Sheets adalah bahasa penulisan yang digunakan untuk mendesai atau mendeskripsikan penampilan dari sebuah *mark-up language*. Salah satu *mark-up language* yang sering digunakan dan identik penggunaannya disandingkan dengan CSS yaitu adalah HTML (Hypertext Markup Language). Penggunaan CSS untuk mendesain sebuah halaman HTML yaitu dengan menyimpan kode CSS dalam file berekstensi .css yang kemudian disematkan pada file HTML yang akan digunakan^[6].

3.7. Javascript

Javascript adalah suatu bahasa kode atau pemrograman yang populer yang digunakan untuk tujuan membuat konten pada website menjadi dinamis, untuk memenuhi tujuan tersebut penggunaan Javascript juga harus disertai penggunaan HTML dan CSS. Beberapa fungsi dari javascript antara lain yaitu membuat situs lebih interaktif dan menarik, dapat berfungsi juga untuk mobile app development, dapat digunakan untuk backend dari sebuah website, tidak hanya untuk frontend^[7].

3.8. Git

Git adalah salah satu tools yang sering digunakan oleh programmer dan developer dalam menjalankan sebuah proyek, dimana git merupakan tools gratis untuk memudahkan dalam mencatat setiap perubahan pada file proyek yang dikerjakan oleh beberapa programmer dan developer^[8].

3.9. Midtrans

Midtrans adalah salah satu layanan payment gateway yang cocok digunakan untuk startup, UMKM dan perusahaan besar. Midtrans dapat menerima pembayaran melalui website, chat hingga SMS^[9].

[Halaman ini sengaja dikosongkan]

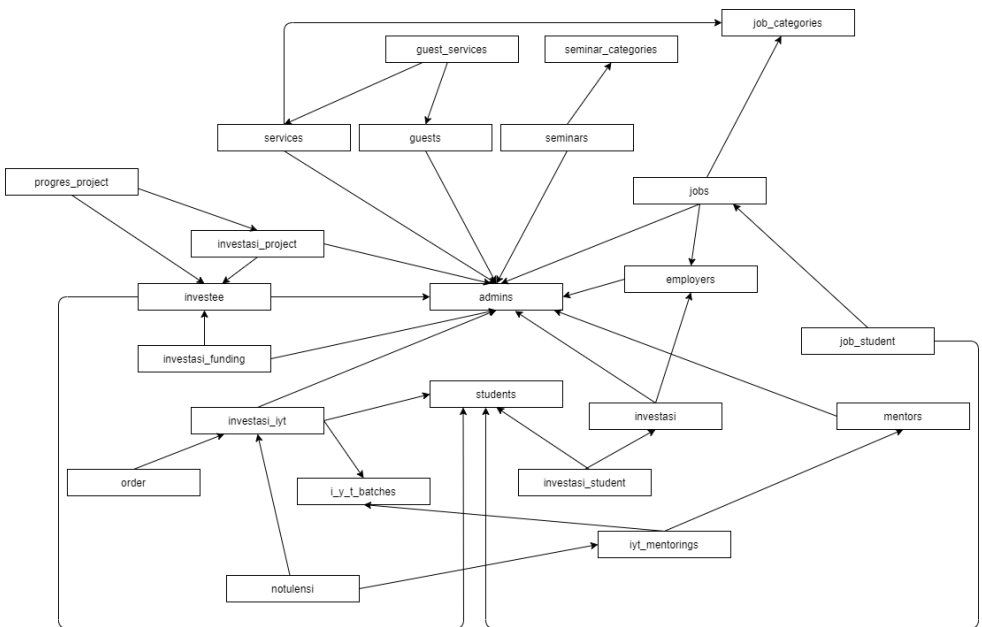
BAB IV

DESAIN SISTEM

Pada bab ini akan menjelaskan desain mengenai sistem yang akan dibangun yaitu untuk Aplikasi Website ITS JobX.

4.1. Desain Data Model

Untuk sistem Aplikasi Website ITS JobX ada beberapa tabel data yang digunakan. Gambar 4.1.1 ini adalah desain data model secara umum

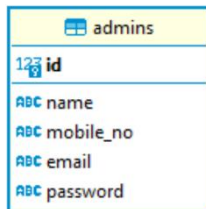


Gambar 4.1.1 Desain Data Model

Berikut ini adalah penjelasan dari tabel-tabel data yang digunakan:

- Tabel admins

Tabel ini menyimpan data user administrator dari sistem, tabel ini akan banyak berelasi dengan tabel lain.



The diagram shows a table named 'admins' with a yellow header. Below the header, the primary key 'id' is listed with a blue key icon and a blue '1' to its left. Below 'id', four attributes are listed: 'name', 'mobile_no', 'email', and 'password', each preceded by a blue 'ABC' icon.

admins	
123	id
ABC	name
ABC	mobile_no
ABC	email
ABC	password

Gambar 4.1.2 Model tabel admins

- Tabel students

Tabel ini menyimpan data user mahasiswa.

students	
123	id
ABC	status
ABC	name
ABC	nrp
ABC	mobile_no
ABC	email
ABC	password
ABC	hobby
ABC	skill
ABC	achievement
ABC	experience
ABC	gender
ABC	birthdate
ABC	address
ABC	city
ABC	province
ABC	berkas_validasi
ABC	created_at
ABC	updated_at

Gambar 4.1.3 Model tabel students

- Tabel guests
Tabel ini menyimpan data user guest.

guests	
123	id
ABC	name
ABC	status
123	status_gs
ABC	pekerjaan
ABC	mobile_no
ABC	email
ABC	password
ABC	berkas_verifikasi

Gambar 4.1.4. Model tabel guests

- Tabel employers

Tabel ini menyimpan data user employer.

employers	
123	id
123	admin_id
123	status
ABC	name
ABC	logo
ABC	website
ABC	address
ABC	city
ABC	province
ABC	fax
ABC	contact_person
ABC	contact_no
ABC	email
ABC	password
ABC	berkas_verifikasi

Gambar 4.1.5 Model tabel employers

- Tabel mentors

Tabel ini menyimpan data user mentors.

mentors	
123	id
123	admin_id
ABC	nip
ABC	name
ABC	mobile_no
ABC	email
ABC	password

Gambar 4.1.6 Model tabel mentors

- Tabel jobs

Tabel ini menyimpan data pekerjaan yang ditawarkan atau dibuat oleh user employer

jobs	
123	id
123	employer_id
123	status
ABC	name
ABC	position
123	job_category_id
ABC	kompesasi
ABC	description
ABC	job_type
ABC	location
123	admin_id
ABC	required_skill
ABC	minimal_qualification
ABC	extra_skill
123	expected_salary_high
123	expected_salary_low
ABC	order_rekrutmen

Gambar 4.1.7 Model tabel jobs

- Tabel seminars

Tabel ini menyimpan data seminar yang dibuat oleh user employer

seminars	
123	id
123	status
ABC	name
ABC	description
ABC	berkas_verifikasi
ABC	poster
ABC	profil_pembicara
ABC	target
ABC	materi
123	fee
🕒	waktu
ABC	location
ABC	contact_person
ABC	contact_no
🕒	created_at
🕒	updated_at
123	employer_id
123	seminar_category_id
123	admin_id

Gambar 4.1.8 Model tabel seminars

- Tabel services

Tabel ini menyimpan data jasa yang dibuat oleh user students atau mahasiswa

services	
123	id
ABC	name
ABC	description
123	status
🕒	created_at
🕒	updated_at
123	student_id
123	job_category_id
123	admin_id

Gambar 4.1.9 Model tabel services

- Tabel `job_categories`

Tabel ini menyimpan data kategori pekerjaan, berelasi one-to-many dengan tabel `jobs` dan tabel `services`

job_categories	
123	id
ABC	name
ABC	slug

Gambar 4.1.10 Model tabel `job_categories`

- Tabel `seminar_categories`

Tabel ini menyimpan data kategori seminar yang berelasi one-to-many dengan tabel `seminars`

seminar_categories	
123	id
ABC	name
ABC	slug
🕒	created_at
🕒	updated_at

Gambar 4.1.11 Model tabel `seminar_categories`

- Tabel `job_student`

Tabel pivot dari tabel `students` dan `jobs` yang mana merupakan hasil relasi many-to-many antara dua tabel tersebut.

job_student	
123	id
123	student_id
123	job_id
ABC	status
ABC	motivation_letter
ABC	cv

Gambar 4.1.12 Model tabel job_student

- Tabel guest_services

Tabel pivot dari tabel guests dan services yang mana merupakan hasil relasi many-to-many antara dua tabel tersebut

guest_services	
123	id
123	service_id
123	guest_id
ABC	status
ABC	status_pekerjaan

Gambar 4.1.13 Model tabel guest_services

- Tabel investee

Tabel yang menyimpan data kelompok usaha mahasiswa yang ingin menghimpun investasi baik investasi project ataupun funding, tabel ini berelasi one-to-many dengan tabel Student

investee	
123	id
ABC	status
123	admin_id
123	student_id
ABC	nama
ABC	address
ABC	city
ABC	province
ABC	fax
ABC	contact_person
ABC	contact_no
ABC	email

Gambar 4.1.14 Model tabel investee

- Tabel investasi_project

Tabel yang menyimpan data investasi project, berelasi one-to-many dengan tabel admins dan investee

investasi_project	
123	id
ABC	nama_investasi
123	investee_id
ABC	status
ABC	status_tempo
ABC	bank
ABC	no_rekening
ABC	atas_nama
ABC	deskripsi_bisnis
123	roi_top
123	roi_bot
123	lembar_total
123	lembar_terbeli
123	harga_saham
🕒	tgl_jatuh_tempo
ABC	berkas_proposal_investasi
ABC	berkas_laporan_keuangan
123	admin_id

Gambar 4.1.15 Model tabel investasi_project

- Tabel investasi_funding

Tabel yang menyimpan data investasi funding, berelasi one-to-many dengan tabel admins dan investee

investasi_funding	
123	id
ABC	nama_investasi
123	investee_id
ABC	status
ABC	status_tempo
123	donasi_target
123	donasi_masuk
ABC	bank
ABC	no_rekening
ABC	atas_nama
ABC	deskripsi_bisnis
123	target
🕒	tgl_jatuh_tempo
ABC	berkas_proposal_investasi
ABC	berkas_laporan_keuangan
123	admin_id

Gambar 4.1.16 Model tabel investasi_funding

- Tabel investasi

Tabel yang menyimpan data investasi yang ditawarkan oleh employer, berelasi one-to-many dengan tabel employer dan admins

investasi	
123	id
ABC	nama_investasi
123	employer_id
ABC	status
ABC	status_tempo
ABC	bank
ABC	no_rekening
ABC	atas_nama
ABC	deskripsi_bisnis
123	roi_top
123	roi_bot
123	lembar_total
123	lembar_terbeli
123	harga_saham
🕒	tgl_jatuh_tempo
ABC	berkas_proposal_investasi
ABC	berkas_laporan_keuangan
123	admin_id

Gambar 4.1.17 Model tabel investasi

- Tabel investasi_student
Tabel pivot untuk menyimpan data pembelian investasi dari mahasiswa untuk investasi employer, hasil relasi many-to-many dari tabel investasi dan student

investasi_student	
123	id
123	student_id
123	investasi_id
ABC	status_bayar
ABC	status_uang_balik
123	lembar_beli
ABC	berkas_ktp
ABC	berkas_bukti_pembayaran
ABC	berkas_payback
ABC	status_payback
🕒	expired_at

Gambar 4.1.18 Model tabel investasi_student

- Tabel progres_project

Tabel yang menyimpan data untuk melaporkan progress dari usaha yang diinvestasikan, berelasi one-to-many dengan tabel investasi_project dan investee

progres_project	
123	id
123	investee_id
123	project_id
🕒	tgl
ABC	berkas_laporan
ABC	deskripsi_laporan
ABC	keterangan_tambahan

Gambar 4.1.19 Model tabel progres_project

- Tabel i_y_t_batches

Tabel yang menyimpan data batch dari IYT

i_y_t_batches	
123	id
123	status
ABC	IYName
ABC	batch
🕒	start_date
🕒	end_date

Gambar 4.1.20 Model tabel i_y_t_batches

- Tabel investasi_iyt

Tabel yang menyimpan data pendaftaran IYT, berelasi one-to-many dengan i_y_t_batches, admins dan students

investasi_iyt	
123	id
ABC	nama_ketua
ABC	invoice_iyt
123	student_id
ABC	status
ABC	nama_kelompok
ABC	berkas_pitch_desk
ABC	berkas_proposal_bisnis
ABC	tahun_masuk
ABC	tahun_keluar
ABC	kategori
123	semester
123	admin_id
123	batch_id

Gambar 4.1.21 Model tabel investasi_iyt

- Tabel iyt_mentorings

Tabel menyimpan jadwal mentoring untuk IYT, berelasi one-to-many dengan tabel i_y_t_batches dan tabel mentors

iyt_mentorings	
123	id
123	batch_id
123	mentor_id
ABC	judul
🕒	tgl_mentoring
ABC	link

Gambar 4.1.22 Model tabel iyt_mentorings

- Tabel notulensi

Tabel pivot menyimpan data notulensi dari setiap Kelompok IYT di setiap mentoring, hasil relasi many-to-many dari tabel iyt_mentorings dan investasi_iyt

notulensi	
123	id
123	iyt_id
123	mentoring_id
ABC	dokumentasi
ABC	notulensi
ABC	komentar

Gambar 4.1.23 Model tabel notulensi

- Tabel order

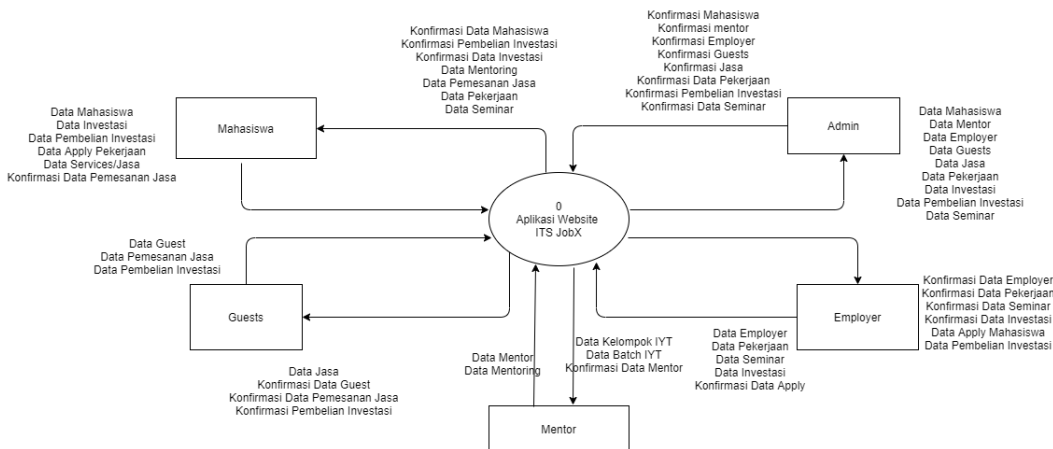
Tabel untuk menyimpan data pembelian investasi, berelasi dengan tabel setiap role students, guests dan berbagai jenis investasi (investasi_project dan investasi_funding)

order	
123	id
ABC	invoice
ABC	nama_investor
ABC	email_investor
ABC	id_investor
ABC	role
ABC	tipe_investasi
ABC	investasi_id
ABC	nama_investasi
ABC	nama_investee
ABC	id_investee
123	lembar_beli
ABC	status
123	total_harga
🕒	order_date
🕒	payment_due
ABC	snap_token

Gambar 4.1.24 Model tabel order

4.2 Desain Data Flow Diagram

Berikut ini adalah desain Data Flow Diagram (DFD) Level 0 untuk sistem aplikasi website ITS JobX, dimana pada gambar 4.2.1 berikut diperlihatkan ada 5 role user utama dalam sistem ini yaitu user mahasiswa, admin, guest, employer dan juga mentor.



Gambar 4.2.1 Data Flow Diagram Level 0

BAB V

IMPLEMENTASI SISTEM

Pada bab ini akan menjelaskan tahap implementasi yang dilakukan untuk pembuatan website ITS JobX.

5.1 Implementasi Source Code

Untuk implementasi source code berikut ini adalah kumpulan code mengenai controller yang merupakan logika utama yang digunakan untuk membangun aplikasi ITS JobX

- Controllers/AdminController.php
Controller yang berisi fungsi-fungsi untuk mengatur user Admin

```
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Hash;
use Session;
use Validator;
use App\Admin;

class AdminController extends Controller
{
    // menampilkan form login admin
    public function showLogin() {
        return view('dashboard.pages.admins.auth.supersecretlogin');
    }

    // login admin
    public function Login(Request $request) {
```

```

$admin = Admin::where('email', '=', $request->input('email'))->first();
if($admin) {
    if(Hash::check($request->input('password'), $admin->password)) {
        $request->session()->put([
            'login'      => true,
            'name'       => $admin->name,
            'id'         => $admin->id,
            'email'      => $admin->email,
            'role'       => 'admin',
            'mobile_no' => $admin->mobile_no,
        ]);
        Session::flash('success', 'Anda berhasil Login');
        return redirect('/dashboard');
    }else{
        // password salah
        Session::flash('error', 'Salah Password');
        return redirect('/admin/login');
    }
}else{
    // admin tidak ditemukan
    Session::flash('error', 'User tidak dapat ditemukan');
    return redirect('/admin/login');
}
}

// menampilkan form register admin
public function showRegister() {
    return view('dashboard.pages.admins.auth.supersecretregister');
}

public function Register(Request $request) {
    $validator = Validator::make($request->all(), [
        'name' => 'required',
        'email' => 'required|email|unique:admins',
    ]);
}

```

```

        'password' => 'required',
        'mobile_no' => 'required'
    ]);

    if ($validator->fails()) {
        Session::flash('error', $validator->errors()->all());
        return redirect()->back()->withInput();
    }

    try {
        Admin::create([
            'name' => $request->input('name'),
            'email' => $request->input('email'),
            'mobile_no' => $request->input('mobile_no'),
            'password' => Hash::make($request->input('password')),
        ]);

        Session::flash('success', 'User berhasil didaftarkan');
        return redirect('/admin/login');
    } catch (\Illuminate\Database\QueryException $e) {
        $errorCode = $e->errorInfo[1];
        if ($errorCode == 1062) {
            return redirect('/admin/register');
        }
    }
}
}

```

Kode 5.1.1. Kode Admin Controller

- Controller/DashboardController.php
Controller yang berisi fungsi-fungsi untuk mengatur Dashboard Admin

```

<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;

```

```

use Session;
use Illuminate\Support\Facades\DB;
use App\Student;
use App\Job;
use App\Employer;
use App\Guest;
use App\Seminar;
use App\Service;
use App\Investasi_project;
use App\Investasi_funding;
use App\Investasi_IYT;
use App\Investee;
use App\IYBatch;

class DashboardController extends Controller
{
    public function getHome(Request $request) {
        if ($request->session()->get('role') == 'admin') {
            return view('dashboard.pages.admins.home');
        } elseif ($request->session()->get('role') == 'employer') {
            return view('dashboard.pages.employer.home');
        } elseif ($request->session()->get('role') == 'student') {
            // * cek apa pernah daftar iyt atau belum
            $id = $request->session()->get('id');
            $IYT = DB::table('investasi_iyt')
                -
            >leftjoin('i_y_t_batches', 'i_y_t_batches.id', 'investasi_iyt.batch_id')
                ->where('i_y_t_batches.status', '=', '1')
                ->where('investasi_iyt.student_id', '=', $id)
                -
            >select('*', 'investasi_iyt.id as iyt_id', 'investasi_iyt.status as status_iyt')
                ->first();

            if (!$IYT) {

```

```

        $IYT = null;
    }

    return view('dashboard.pages.student.home')->with('iyt', $IYT);

} else if ($request->session()->get('role') == 'guest') {
    return view ('dashboard.pages.guest.home');
} else if ($request->session()->get('role') == 'mentor') {
    return view ('dashboard.pages.mentor.home');
}

return view('/');
}

public function getInvestee() {
    return view('dashboard.pages.investee.home');
}

// manage users admin
public function getNewStudents()
{
    $students = Student::where('students.status', '0')
        ->select('students.*')
        ->paginate(20);

    return view ('dashboard.pages.admins.newstudents')->with('students', $students);
}

public function approveNewStudents(Request $request, $id)
{
    $student = Student::find($id);
    $student->status = 1;
    // $student->admin_id = $request->session()->get('id');
    $student->save();
    return redirect()->back();
}

public function rejectNewStudents(Request $request, $id)

```



```

{
    $student = Student::find($id);
    $student->status = 2;
    // $student->admin_id = $request->session()->get('id');
    $student->save();
    return redirect()->back();
}

public function getNewServices()
{
    $services = Service::where('services.status', '0')
        ->leftjoin('students', 'students.id', 'services.student_id')
        -
>leftjoin('job_categories', 'job_categories.id', 'services.job_category_id')
        -
>select('services.*', 'students.name as stdname', 'job_categories.name as category_name')
        ->paginate(20);
    return view('dashboard.pages.admins.newservices')-
>with('services', $services);
}

public function getApprovedServices()
{
    $services = Service::where('services.status', '1')
        ->leftjoin('students', 'students.id', 'services.student_id')
        -
>leftjoin('job_categories', 'job_categories.id', 'services.job_category_id')
        -
>select('services.*', 'students.name as stdname', 'job_categories.name as category_name')
        ->paginate(20);
    return view('dashboard.pages.admins.approvedservices')-
>with('services', $services);
}

```

```

    }

    public function getUnapprovedServices()
    {
        $services = Service::where('services.status', '2')
            ->leftjoin('students', 'students.id', 'services.student_id')
            -
        >leftjoin('job_categories', 'job_categories.id', 'services.job_category_id')
            -
        >select('services.*', 'students.name as stdname', 'job_categories.name as category_name')

            ->paginate(20);

        return view('dashboard.pages.admins.unapprovedservices')-
        >with('services', $services);
    }

    public function approveNewServices(Request $request, $id)
    {
        $service = Service::find($id);

        $service->status = 1;

        $service->admin_id = $request->session()->get('id');

        $service->save();

        return redirect()->back();
    }

    public function rejectNewServices(Request $request, $id)
    {
        $service = Service::find($id);

        $service->status = 2;

        $service->admin_id = $request->session()->get('id');

        $service->save();

        return redirect()->back();
    }

    //manage employers guest

    public function getNewGuests()

```

```

{
    $guests = Guest::where('status_gs', '0')->paginate(20);
    return view ('dashboard.pages.admins.newguests')-
>with('guests', $guests);
}

public function approveNewGuests(Request $request, $id)
{
    $guest = Guest::find($id);
    $guest->status_gs = 1;
    $guest->save();
    return redirect()->back();
}

public function rejectNewGuests(Request $request, $id)
{
    $guest = Guest::find($id);
    $guest->status_gs = 2;
    $guest->save();
    return redirect()->back();
}

public function deleteGuests($id)
{
    $guest = Guest::find($id);
    $guest->delete();
    return redirect()->back();
}

//manage employers admin
public function getNewEmployers()
{
    $employers = Employer::where('status', '0')->paginate(20);
    return view ('dashboard.pages.admins.newemployers')-
>with('employers', $employers);
}

```

```

public function approveNewEmployers(Request $request, $id)
{
    $employer = Employer::find($id);
    $employer->status = 1;
    $employer->admin_id = $request->session()->get('id');
    $employer->save();
    return redirect()->back();
}

public function rejectNewEmployers(Request $request, $id)
{
    $employer = Employer::find($id);
    $employer->status = 2;
    $employer->admin_id = $request->session()->get('id');
    $employer->save();
    return redirect()->back();
}

public function getNewJobs() {
    $jobs = Job::where('jobs.status', '0')
        ->leftjoin('employers', 'employers.id', 'employer_id')
        -
    >leftjoin('job_categories', 'job_categories.id', 'job_category_id')
        -
    >select('jobs.*', 'employers.name as employer_name', 'job_categories.name as cate
gory_name')

        ->paginate(20);

    return view ('dashboard.pages.admins.newjobs')->with('jobs', $jobs);
}

public function approveNewJobs(Request $request, $id)
{
    $job = Job::find($id);
    $job->status = 1;
    $job->admin_id = $request->session()->get('id');

```

```

        $job->save();
        return redirect()->back();
    }

    public function rejectNewJobs(Request $request, $id)
    {
        $job = Job::find($id);
        $job->status = 2;
        $job->admin_id = $request->session()->get('id');
        $job->save();
        return redirect()->back();
    } // manage jobs admin

    public function getNewSeminars() {
        $seminars = Seminar::where('status', '0')->paginate(20);
        return view('dashboard.pages.admins.newseminars')-
>with('seminars', $seminars);
    }

    public function approveNewSeminars(Request $request, $id)
    {
        $seminar = Seminar::find($id);
        $seminar->status = 1;
        $seminar->admin_id = $request->session()->get('id');
        $seminar->save();
        return redirect()->back();
    }

    public function rejectNewSeminars(Request $request, $id)
    {
        $seminar = Seminar::find($id);
        $seminar->status = 2;
        $seminar->admin_id = $request->session()->get('id');
        $seminar->save();
        return redirect()->back();
    }

    public function getNewProjInvestment() {

```

```

        $investasi = Investasi_project::where('investasi_project.status', '0')
            ->leftjoin('investee', 'investee.id', 'investee_id')
            ->leftjoin('students', 'students.id', 'investee.student_id')
            -
>select('investasi_project.*', 'students.name as mahasiswa', 'investee.nama as in
vestee', 'investee.contact_person as cp', 'investee.contact_no as kontak')
            ->paginate(20);

        return view('dashboard.pages.admins.newprojinvestment')-
>with('investasi', $investasi);
    }

    public function approveNewProjInvestment(Request $request, $id)
    {
        $investasi = Investasi_project::find($id);
        $investasi->status = 1;
        $investasi->status_tempo = 1;
        $investasi->admin_id = $request->session()->get('id');
        $investasi->save();
        return redirect()->back();
    }

    public function rejectNewProjInvestment(Request $request, $id)
    {
        $investasi = Investasi_project::find($id);
        $investasi->status = 2;
        $investasi->admin_id = $request->session()->get('id');
        $investasi->save();
        return redirect()->back();
    }

    public function downloadproposalproj($proposal)
    {
        $where = [
            'investasi_project.id' => $proposal,
        ];

```

```

$berkas_db = DB::table('investasi_project')
->select('investasi_project.berkas_proposal_investasi as berkas')
->where($where)
->first();

$file = public_path('data_files/Student/Investee/Project/proposal_investasi/'.$berkas_db->berkas);

return response()->download($file, $berkas_db->berkas);
}

public function getNewFundInvestment() {
    $investasi = Investasi_funding::where('investasi_funding.status', '0')
        ->leftjoin('investee', 'investee.id', 'investee_id')
        ->leftjoin('students', 'students.id', 'investee.student_id')
        -
>select('investasi_funding.*', 'students.name as mahasiswa', 'investee.nama as investee', 'investee.contact_person as cp', 'investee.contact_no as kontak')
        ->paginate(20);

    return view('dashboard.pages.admins.newfundinvestment')-
>with('investasi', $investasi);
}

public function approveNewFundInvestment(Request $request, $id)
{
    $investasi = Investasi_funding::find($id);
    $investasi->status = 1;
    $investasi->status_tempo = 1;
    $investasi->admin_id = $request->session()->get('id');
    $investasi->save();
    return redirect()->back();
}

public function rejectNewFundInvestment(Request $request, $id)
{
    $investasi = Investasi_funding::find($id);
    $investasi->status = 2;

```

```

        $investasi->admin_id = $request->session()->get('id');
        $investasi->save();
        return redirect()->back();
    }

    public function getNewInvestees()
    {
        $investees = Investee::where('investee.status', '0')
            ->leftjoin('students', 'students.id', 'investee.student_id')
            ->select('investee.*', 'students.name as nama_mhs')
            ->paginate(20);

        return view('dashboard.pages.admins.newinvestees')-
>with('investees', $investees);
    }

    public function approveNewInvestees(Request $request, $id)
    {
        $investee = Investee::find($id);
        $investee->status = 1;
        $investee->admin_id = $request->session()->get('id');
        $investee->save();
        return redirect()->back();
    }

    public function rejectNewInvestees(Request $request, $id)
    {
        $investee = Investee::find($id);
        $investee->status = 1;
        $investee->admin_id = $request->session()->get('id');
        $investee->save();
        return redirect()->back();
    }

    public function getListAllIYT() {
        $batch = DB::table('i_y_t_batches')

```



```

        ->select('id')
        ->where('status','=', 1 )
        ->first();

$iyts= DB::table('investasi_iyt')
        ->join('students', 'students.id', '=', 'investasi_iyt.student_id')
        ->select('investasi_iyt.*', 'i_y_t_batches.batch')
        ->where('investasi_iyt.status','=', '0')
        ->where('batch_id','=', $batch->id)
        ->select('investasi_iyt.*', 'students.email')
        ->paginate(20);

// $batch= IYBatch::where('id',$iyts)

return view('dashboard.pages.admins.iyt-list-all')->with('iyts', $iyts);
}

public function approveIYT(Request $request, $id)
{
    $investasi = Investasi_IYT::find($id);
    $investasi->status = 1;
    $investasi->admin_id = $request->session()->get('id');
    $investasi->save();
    return redirect()->back();
}

public function rejectIYT(Request $request, $id)
{
    $investasi = Investasi_IYT::find($id);
    $investasi->status = 0;
    $investasi->admin_id = $request->session()->get('id');
    $investasi->save();
    return redirect()->back();
}

public function getListQualifyIYT()
{
    $batch = DB::table('i_y_t_batches')

```

```

        ->select('id')
        ->where('status','=', 1 )
        ->first();

$iyts= DB::table('investasi_iyt')
        ->join('students', 'students.id', '=', 'investasi_iyt.student_id')
        ->select('investasi_iyt.*', 'i_y_t_batches.batch')
        ->where('investasi_iyt.status','=', '1')
        ->where('batch_id','=', $batch->id)
        ->select('investasi_iyt.*', 'students.email')
        ->paginate(20);

return view('dashboard.pages.admins.iyt-qualify')->with('iyts', $iyts);
}

public function viewCreateIYBatch(){
    return view('dashboard.pages.admins.iyt-create-batch');
}

public function viewEditIYBatch($id){
    $batch = DB::table('i_y_t_batches')
        ->where('id','=', $id)
        ->first();

    $startDate = \Carbon\Carbon::parse($batch->start_date)->format('d-m-Y');
    $endDate = \Carbon\Carbon::parse($batch->end_date)->format('d-m-Y');
    // dd($startDate, $endDate);

    return view('dashboard.pages.admins.edit-batch')->with('batch', $batch)-
>with('startDate', $startDate)->with('endDate', $endDate);
}

public function createIYBatch(Request $request){
    $this->validate($request, [
        'batch' => 'required|numeric|unique:i_y_t_batches',
        'start_date' => 'required|date',
        'end_date' => 'required|date|afteron_equal:start_date',
    ]);
}

```

```

        $startDate = \Carbon\Carbon::parse($request->input('start_date'))-
>format('Y-m-d');

        $endDate = \Carbon\Carbon::parse($request->input('end_date'))->format('Y-
m-d');

        // $formatDate = \Carbon\Carbon::parse($request->input('waktu'))-
>format('Y-m-d');

        $batch = DB::table('i_y_t_batches')
            ->where('status','=', '1')
            ->first();

        try{
            if($batch)
            {
                IYBatch::create([
                    'name'          => $request->input('IYName'),
                    'batch'         => $request->input('batch'),
                    'start_date'     => $startDate,
                    'status'        => 0,
                    'end_date'      => $endDate,
                ]);
            }
            else
            {
                IYBatch::create([
                    'name'          => $request->input('IYName'),
                    'batch'         => $request->input('batch'),
                    'start_date'     => $startDate,
                    'status'        => 1,
                    'end_date'      => $endDate,
                ]);
            }

            Session::flash('success', 'Batch IYT Baru telah berhasil dibuat');
            return view('dashboard.pages.admins.iyt-create-batch');
        }
    }
}

```

```

    }

    catch(\Illuminate\Database\QueryException $e)
    {
        $errorCode = $e->errorInfo[1];
        $errorMsg = $e->errorInfo[2];
        if ($errorCode == 1062) {
            return redirect('/');
        }
        Session::flash('error', $errorMsg);
        return view('dashboard.pages.admins.iyt-create-batch');
    }
}

public function EditIYBatch(Request $request, $id){
    $this->validate($request, [
        'start_date'      => 'required|date',
        'end_date'        => 'required|date|after_or_equal:start_date',
    ]);

    $startDate = \Carbon\Carbon::parse($request->input('start_date'))->format('Y-m-d');
    $endDate = \Carbon\Carbon::parse($request->input('end_date'))->format('Y-m-d');

    $batches = IYBatch::find($id);
    $batches->start_date = $startDate;
    $batches->end_date = $endDate;
    $batches->save();
    return redirect('admin/list-batches');
}

public function getListBatches()
{
    $batches = DB::table('i_y_t_batches')->paginate(20);
    return view('dashboard.pages.admins.list-batch')->with('batches', $batches);
}

```

```
}  
}
```

Kode 5.1.2. Kode Dashboard Controller

- Controllers/DashboardEmployerController.php
Controller yang berisi fungsi-fungsi untuk mengatur Dashboard Employer

```
<?php  
namespace App\Http\Controllers;  
use App\Http\Controllers\Controller;  
use Illuminate\Http\Request;  
use Illuminate\Support\Facades\DB;  
use Session;  
use Validator;  
use App\Student;  
use App\Job;  
use App\Employer;  
use App\Investasi;  
use App\InvestasiStudent;  
use App\Seminar;  
class DashboardEmployerController extends Controller  
{  
    public function acceptNewApplicants(Request $request, $id)  
    {  
        $acc = DB::table('job_student')  
            ->where('id', $id)  
            ->update([  
                'status' => 1,  
                'updated_at' => \Carbon\Carbon::now(),  
            ]);  
        return redirect()->back();  
    }  
}
```

```

public function rejectNewApplicants(Request $request, $id)
{
    $acc = DB::table('job_student')
        ->where('id', $id)
        ->update([
            'status' => 2,
            'updated_at' => \Carbon\Carbon::now(),
        ]);

    return redirect()->back();
}

public function getCreateSeminar()
{
    return view('dashboard.pages.employer.create-seminar');
}

public function postCreateSeminar(Request $request)
{
    $this->validate($request, [
        'name' => 'required|max:255',
        'location' => 'required|max:255',
        'category' => 'required',
        'description' => 'required|max:255',
        'contact_person'=> 'required|max:255',
        'contact_no' => 'required|max:14',
        'fee' => 'required',
        // 'berkas_sewa' => 'required|mimes:pdf|max:2048',
        'profil_pemb' => 'required|mimes:pdf|max:2048',
        'waktu' => 'required|date',
        'target'=> 'required|max:255',
        'materi'=> 'required|max:255',
        'poster' => 'required|image|mimes:jpeg,png,jpg,gif,svg|max:4096'
    ]);

    $nama= str_replace(' ','_',$request->input('name'));

```

```

        $location = str_replace(' ','_',$request->input('location'));
        $desc= md5($request->input('description'));
        $berkas_profil= $request->file('profil_pemb');
        $extension_profil= $berkas_profil->getClientOriginalExtension();
        $filenama= 'Profil-
pembicara_'.$nama.'_'. '_'.$location.'_'. '$desc.'. '$extension_profil';
        $tujuannya = 'data_files/Employer/Seminar/Profil pembicara';
        $berkas_profil->move($tujuannya,$filenama);
        $fileposter = $request->file('poster');
        $extension_poster= $fileposter->getClientOriginalExtension();
        $namafile= 'Poster_'.$nama.'_'. '_'.$location.'_'. '$desc.'. '$extension
n_poster;
        $tujuan_upload = 'data_files/Employer/Seminar/Poster';
        $fileposter->move($tujuan_upload,$namafile);
        $formatDate = \Carbon\Carbon::parse($request->input('waktu'))-
>format('Y-m-d');
        try{
            Seminar::create([
                'name' => $request->input('name'),
                'location' => $request->input('location'),
                'seminar_category_id' => $request->input('category'),
                'description' => $request->input('description'),
                'contact_person' => $request->input('contact_person'),
                'contact_no' => $request->input('contact_no'),
                'fee' => $request->input('fee'),
                'employer_id' => $request->session()->get('id'),
                // 'berkas_verifikasi' => $filename,
                'waktu' => $formatDate,
                'target' => $request->input('target'),
                'materi' => $request->input('materi'),
                'profil_pembicara' => $filenama,
                'poster' => $namafile,
            ]);
        } catch (Exception $e) {
            //
        }
    }
}

```

```

    ]);
    Session::flash('success', 'Seminar berhasil didaftarkan, silahkan tun
    ggu konfirmasi seminar');

    return view('dashboard.pages.employer.create-seminar');
}

catch(\Illuminate\Database\QueryException $e)
{
    $errorCode = $e->errorInfo[1];
    $errorMsg = $e->errorInfo[2];
    if ($errorCode == 1062) {
        return redirect('/');
    }

    Session::flash('error', $errorMsg);
    return view('dashboard.pages.employer.create-seminar');
}
}

public function getCreateJob()
{
    return view('dashboard.pages.employer.create-job');
}

public function postCreateJob(Request $request)
{
    // dd($request);

    $this->validate($request, [
        'name' => 'required|max:255',
        'category' => 'required',
        'job_type' => 'required',
        'position' => 'required|max:255',
        'location' => 'required|max:255',
        'description' => 'required',
        'minimal_qualification' => 'required|max:255',
        'required_skill' => 'required|max:255',
    ]);
}

```



```

        'extra_skill'           => 'max:255',
        'expected_salary_high' => 'required|max:11',
        'expected_salary_low'  => 'required|max:11',
        'kompensasi'           => 'max:255',
        'order_rekrutmen'      => 'required|mimes:pdf|max:2048',
    ]);

    if (is_null($request->input('extra_skill'))){
        $extra_skill = '-';
    }else {
        $extra_skill = $request->input('extra_skill');
    }

    try {
        $berkas= $request->file('order_rekrutmen');
        $namaemp= str_replace(' ','_',$request->session()->get('name'));
        $nama= str_replace(' ','_',$request->input('name'));
        $extension= $berkas->getClientOriginalExtension();
        $filename= $namaemp.'_'.$nama.'.'.$extension;
        $tujuan = 'data_files/Employer/Job/Order Rekrutmen';
        $berkas->move($tujuan,$filename);

        Job::create([
            'name'           => $request->input('name'),
            'employer_id'    => $request->session()->get('id'),
            'job_category_id' => $request->input('category'),
            'job_type'        => $request->input('job_type'),
            'position'        => $request->input('position'),
            'location'        => $request->input('location'),
            'description'     => $request->input('description'),
            'minimal_qualification' => $request->
>input('minimal_qualification'),
            'required_skill' => $request->input('required_skill'),
            'extra_skill'    => $extra_skill,
        ]);
    } catch (Exception $e) {
        //
    }
}

```

```

        'expected_salary_high' => $request->input('expected_salary_high'),
        'expected_salary_low' => $request->input('expected_salary_low'),
        'kompesasi' => $request->input('kompensasi'), //typo di tabel, males benerin
        'order_rekrutmen' => $filename,
    ]);
    Session::flash('success', 'Job berhasil didaftarkan, silahkan tunggu konfirmasi job');
    return view('dashboard.pages.employer.create-job');
}
catch(\Illuminate\Database\QueryException $e)
{
    $errorCode = $e->errorInfo[1];
    $errorMsg = $e->errorInfo[2];
    if ($errorCode == 1062) {
        return redirect('/');
    }
    Session::flash('error', $errorMsg);
    return view('dashboard.pages.employer.create-job');
}
// return view('dashboard.pages.employer.create-job');
}
public function getCreateInvestation()
{
    return view('dashboard.pages.employer.create-investastion');
}
public function postCreateInvestation(Request $request){
    $input = $request->all();
    $validator = Validator::make($input, [
        'namainvestasi' => 'required|max:191',
        'description' => 'required',
    ]);
    if ($validator->fails()) {
        Session::flash('error', $validator->errors->first());
        return view('dashboard.pages.employer.create-investastion');
    }
    $data = [
        'order_rekrutmen' => $request->input('order_rekrutmen'),
        'description' => $request->input('description'),
        'namainvestasi' => $request->input('namainvestasi'),
        'expected_salary_high' => $request->input('expected_salary_high'),
        'expected_salary_low' => $request->input('expected_salary_low'),
        'kompesasi' => $request->input('kompensasi'),
    ];
    $job = new Job($data);
    $job->save();
    Session::flash('success', 'Job berhasil didaftarkan, silahkan tunggu konfirmasi job');
    return view('dashboard.pages.employer.create-job');
}
}

```

```

        'namabank'      => 'required|max:191',
        'nomorrekening' => 'required|numeric',
        'atasnama'      => 'required|max:191',
        'hargaperlembar' => 'required|numeric',
        'totallembar'   => 'required|numeric|min: 1',
        'roi_d'         => 'required|numeric',
        'roi_u'         => 'required|numeric|gt:roi_d',
        'duedate'       => 'required|date',
        'proposalinvestasi' => 'required|mimes:pdf|max:2048',
        'laporankeuangan' => 'required|mimes:pdf|max:2048',
    ]);

    if ($validator->fails()) {
        Session::flash('error', $validator->errors()->first());
        return redirect()->back()->withInput();
    }

    $berkasinvestasi= $request->file('proposalinvestasi');
    $berkaskeuangan= $request->file('laporankeuangan');
    $employerid= $request->session()->get('id');
    $employername = $request->session()->get('name');
    $tujuaninv = 'data_files/proposal_investasi';
    $tujuankeu = 'data_files/lap_keu';
    $extension= 'pdf';

    $filenameinv= "Invesment_". $employerid.md5('_INV_'. $employername.'_'.
$request->input('description')).'.'. $extension;

    $filenamekeu= "Keuangan_". $employerid.md5('_KEU_'. $employername.'_'. $
request->input('description')).'.'. $extension;

    $berkasinvestasi->move($tujuaninv,$filenameinv);
    $berkaskeuangan->move($tujuankeu,$filenamekeu);

    $formatDate = \Carbon\Carbon::parse($request->input('duedate'))->format('Y-m-d');

    try{
        Investasi::create([

```

```

        'nama_investasi'      => $request->input('namainvestasi'),
        'employer_id'        => $request->session()->get('id'),
        'status'              => 0,
        'status_tempo'        => 0,
        'bank'                 => $request->input('namabank'),
        'no_rekening'          => $request->input('nomorrekening'),
        'atas_nama'            => $request->input('atasnama'),
        'deskripsi_bisnis'     => $request->input('description'),
        'roi_top'               => $request->input('roi_u'),
        'roi_bot'               => $request->input('roi_d'),
        'lembar_total'          => $request->input('totallembar'),
        'lembar_terbeli'        => 0,
        'harga_saham'           => $request->input('hargaperlembar'),
        'tgl_jatuh_tempo'       => $formatDate,
        'berkas_proposal_investasi' => $filenameinv,
        'berkas_laporan_keuangan' => $filenamekeu,
    ];

    Session::flash('success', 'Investasi berhasil didaftarkan, silahkan t
unggu konfirmasi dari pihak admin');

    return view('dashboard.pages.employer.create-investastion');
}

catch(\Illuminate\Database\QueryException $e)
{
    $errorCode = $e->errorInfo[1];
    $errorMsg = $e->errorInfo[2];
    if ($errorCode == 1062) {
        return redirect('/');
    }
    Session::flash('error', $errorMsg);
    return view('dashboard.pages.employer.create-investastion');
}
}

```

```

public function getInvestationsApproval(Request $request)
{
    $where = [
        'employer_id' => $request->session()->get('id'),
    ];
    $investment = Investasi::where($where)->paginate(20);
    // dd($seminars);
    return view('dashboard.pages.employer.investment-approval')-
>with('investations', $investment);
}

public function getPaidInvestor(Request $request)
{
    $where = [
        'employer_id' => $request->session()->get('id'),
        'status_bayar' =>1,
    ];
    $where2 = [
        'employer_id' => $request->session()->get('id'),
    ];
    $investations = DB::table('investasi_student')
        -
    >join('investasi', 'investasi_student.investasi_id', 'investasi.id')
        -
    >join('employers', 'investasi.employer_id', 'employers.id')
        -
    >join('students', 'investasi_student.student_id', 'students.id')
        ->select(
            'investasi_student.id as isid',
            'investasi_student.student_id as isstudent_id'
        ,
            'investasi_student.investasi_id as isinvestas
i_id',

```

```

        'investasi_student.status_bayar as isstatus_b
ayan',
        'investasi_student.status_uang_balik as issta
tus_uang_balik',
        'investasi_student.lembar_beli as islembar_be
li',
        'investasi_student.berkas_ktp as isktp_invest
or',
        'investasi_student.berkas_bukti_pembayaran as
isbukti_bayar_investor',
        'investasi_student.updated_at as updated',
        'students.name as s_name',
        'students.id as s_id',
        'students.nrp as nrp',
        'students.email as s_email',
        'students.mobile_no as s_mobile_no',
        'students.skill as s_skill',
        'students.achievment as s_achievment',
        'students.experience as s_experience',
        'students.city as s_city',
        'students.province as s_province',
        'investasi.employer_id as employer_id',
        'investasi.nama_investasi as in_nama_investas
i')

->where($where)
->paginate(20);

$employerInvestation = DB::table('investasi')
-
>select('investasi.id','investasi.nama_investasi')
->where($where2)
->get();

```

```

        return view('dashboard.pages.employer.investasion-paid-investor', compact('investations','employerInvestation'));
    }

    public function confirmPaidInvestor(Request $request, $id){
        $invid = DB::table('investasi_student')
            ->select('investasi_student.investasi_id as investasi_id',
                    'investasi_student.lembar_beli as beli')
            ->where('id', $id)
            ->get();

        $currentL = DB::table('investasi')
            ->select('investasi.lembar_terbeli as terbeli')
            ->where('id', $invid->first()->investasi_id)
            ->get();

        $updateL=$currentL->first()->terbeli + $invid->first()->beli;
        $acc = DB::table('investasi_student')
            ->where('id', $id)
            ->update([
                'status_bayar' => 2,
                'updated_at' => \Carbon\Carbon::now(),
            ]);

        $acc2 = DB::table('investasi')
            ->where('id', $invid->first()->investasi_id)
            ->update([
                'lembar_terbeli' => $updateL,
                'updated_at' => \Carbon\Carbon::now(),
            ]);

        return redirect()->back();
    }

    public function getDueInvestation(Request $request){
        $now= \Carbon\Carbon::now();
        $investation = DB::table('investasi')
            ->where('employer_id', $request->session()->get('id'))
    }

```

```

        ->Where('tgl_jatuh_tempo','<',$now)
        ->Where('status_tempo',2)
        ->paginate(20);

        return view('dashboard.pages.employer.investastion-due-payments')-
>with('investastions', $investastion);
    }

    public function getDueInvestastionInvestor(Request $request,$id){
        $investastion = DB::table('investasi_student')
            -
>join('students', 'investasi_student.student_id', 'students.id')
            -
>join('investasi', 'investasi_student.investasi_id', 'investasi.id')
            ->select(
                'investasi_student.id as isid',
                'investasi_student.student_id as isstudent_id',
                'investasi_student.investasi_id as isinvestasi_id',
                'investasi_student.status_bayar as isstatus_bayar',
                'investasi_student.status_uang_balik as isstatus_uang
_balik',

                'investasi_student.lembar_beli as islembar_beli',
                'investasi_student.berkas_ktp as isktp_investor',
                'investasi_student.berkas_bukti_pembayaran as isbukti
_bayar_investor',

                'investasi_student.updated_at as updated',
                'investasi_student.status_payback as status_payback',
                'students.name as s_name',
                'students.id as s_id',
                'students.nrp as nrp',
                'students.email as s_email',
                'students.mobile_no as s_mobile_no',
                'students.skill as s_skill',
                'students.achievement as s_achievement',

```



```

        'students.experience as s_experience',
        'students.city as s_city',
        'students.province as s_province',
        'investasi.employer_id as employer_id',
        'investasi.nama_investasi as in_nama_investasi')
->where('investasi_id', $id)
->Where('status_bayar', '=', 2)
->paginate(20);

return view('dashboard.pages.employer.investastion-due-payments-
investor')->with('investations', $investment);
}

public function saveBuktiPembayaran(Request $request,$isid){
    // dd($isid);
    $input = $request->all();
    $validator = Validator::make($input, [
        'payback' => 'required|mimes:pdf|max:2048',
    ]);
    if ($validator->fails()) {
        Session::flash('error', $validator->errors()->first());
        return redirect()->back()->withInput();
    }
    $berkaspayback= $request->file('payback');
    $current_date_time = \Carbon\Carbon::now()->toDateTimeString();
    $studentid= $request->session()->get('id');
    $studentname= $request->session()->get('name');
    $extension= '.pdf';
    $tujuan= 'data_files/berkas_bukti/'.$isid;
    $nama= "payback_".md5($studentid.$studentname.$isid.$current_date_time).$
extension;

    try {
        $berkaspayback->move($tujuan,$nama);
    } catch(\Illuminate\Database\QueryException $e)

```

```

{
    $errorCode = $e->errorInfo[1];
    $errorMsg = $e->errorInfo[2];
    if ($errorCode == 1062) {
        return redirect('/');
    }
    Session::flash('error', $errorMsg);
    return redirect()->back();
}

// dd($isid);

$acc = DB::table('investasi_student')
    ->where('id', $isid)
    ->update([
        'berkas_payback' => $nama,
        'status_payback' => 1,
    ]);
return redirect()->back();
}
}

```

Kode 5.1.3. Kode Dashboard Employer Controller

- **Controllers/DashboardGuestController.php**
Controller yang berisi fungsi-fungsi untuk mengatur Dashboard Guest

```

<?php
namespace App\Http\Controllers;
use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;
use Session;
use App\Order;
class DashboardGuestController extends Controller

```

```

{
    public function getListJasa(Request $request)
    {
        $where = [
            'guest_id' => $request->session()->get('id'),
            'guest_services.status'=>0,
        ];
        $services = DB::table('guest_services')
            ->join('guests', 'guest_services.guest_id', 'guests.id')
            -
        >join('services', 'guest_services.service_id', 'services.id')
            ->join('students', 'services.student_id', 'students.id')
            ->where($where)
            -
        >select('guest_services.id as pivot_id', 'services.id as id', 'guest_services.stat
us_pekerjaan', 'guest_services.updated_at', 'students.name as stdname', 'services
.name as servname', 'guest_services.status')
            ->paginate(20);

        return view('dashboard.pages.guest.list-jasa')-
    >with('services', $services);
    }

    public function getListJasaDone(Request $request)
    {
        $where = [
            'guest_id' => $request->session()->get('id'),
            'guest_services.status_pekerjaan'=>1,
            'guest_services.status'=>1,
        ];
        $services = DB::table('guest_services')
            ->join('guests', 'guest_services.guest_id', 'guests.id')

```

```

-
>join('services', 'guest_services.service_id', 'services.id')
    ->join('students', 'services.student_id', 'students.id')
    ->where($where)
-

>select('guest_services.id as pivot_id', 'services.id as id','guest_services.stat
us_pekerjaan', 'guest_services.updated_at', 'students.name as stdname', 'services
.name as servname', 'guest_services.status')
    ->paginate(20);

    return view('dashboard.pages.guest.list-jasa-done')-
>with('services', $services);
}

public function getListJasaInProgress(Request $request)
{
    $where = [
        'guest_id' => $request->session()->get('id'),
        'guest_services.status_pekerjaan'=>0,
        'guest_services.status'=>1,
    ];

    $services = DB::table('guest_services')
        ->join('guests', 'guest_services.guest_id', 'guests.id')
        -
>join('services', 'guest_services.service_id', 'services.id')
    ->join('students', 'services.student_id', 'students.id')
    ->where($where)
    -

>select('guest_services.id as pivot_id', 'services.id as id','guest_services.stat
us_pekerjaan', 'guest_services.updated_at', 'students.name as stdname', 'services
.name as servname', 'guest_services.status')
    ->paginate(20);

    return view('dashboard.pages.guest.list-jasa-inprogress')-
>with('services', $services);

```

```

    }

    public function getOrderList(Request $request)
    {
        $where = [
            'id_investor' => $request->session()->get('id'),
            'role' => $request->session()->get('role'),
        ];

        $orders = Order::where($where)->orderBy('updated_at', 'desc')-
>paginate(25);

        return view('dashboard.pages.order-list')->with('orders', $orders);
    }

    public function showProjectListStudent(Request $request)
    {
        $guestid= $request->session()->get('id');
        $investment = DB::table('order')
            ->select('order.*')
            ->where('order.id_investor', '=', $guestid)
            ->where('role', '=', 'guest')
            ->where('status', '=', 'paid')
            ->where('tipe_investasi', '=', 'project')
            ->paginate(25);

        return view('dashboard.pages.guest.on-going-project-list')-
>with('investment', $investment);
    }

    public function showDetailInvestment(Request $request, $id)
    {
        $guestid= $request->session()->get('id');
        $order = DB::table('order')
            ->select('order.id_investee')
            ->where('id_investor', '=', $guestid)
            ->where('status', '=', 'paid')
            ->where('role', '=', 'guest')

```

```

        ->where('id_investasi', '=', $id)
        ->where('tipe_investasi', '=', 'project')
        ->first();

    $investee = DB::table('investee')
        ->select('investee.*')
        ->where('id', '=', $order->id_investee)
        ->first();

    $investment = DB::table('investasi_project')
        ->select('investasi_project.*')
        ->where('investasi_project.id', '=', $id)
        ->first();

    $listprogres = DB::table('progres_project')
        -> select('progres_project.*')
        -> where('progres_project.project_id', '=', $id)
        -> orderByRaw('tgl DESC')
        ->paginate(8);

    return view('dashboard.pages.guest.detail-
investment',compact('investee','listprogres','investment'));
}
}

```

Kode 5.1.4. Kode Dashboard Guest Controller

- **Controllers/DashboardInvesteeController.php**
Controller yang berisi fungsi-fungsi untuk mengatur Dashboard Investee

```

<?php
namespace App\Http\Controllers;
use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;
use Validator;
use App\Investee;

```

```

use App\Investasi_project;
use App\Investasi_funding;
use App\Investasi_IYT;
use App\Student;
use Session;

class DashboardInvesteeController extends Controller
{
    public function showRegister(Request $request)
    {
        $id = $request->session()->get('id');
        $student = DB::table('students')->find($id);
        return view('dashboard.pages.student.register-investee')-
>with('student', $student);
    }

    public function getRegisterStatus(Request $request)
    {
        $id = $request->session()->get('id');
        $investee = DB::table('investee')->where('student_id', '=', $id)-
>first();
        if ($investee)
            return view('dashboard.pages.student.register-investee-status')-
>with('investee', $investee);
        else
            return redirect('dashboard/register-investee');
    }

    public function registerNew(Request $request)
    {
        // dd($request);
        $id = $request->session()->get('id');
        $student = DB::table('students')->find($id);
        $investeeCheck = DB::table('investee')->where('student_id', '=', $id)-
>first();

```

```

if ($investeeCheck){
    Session::flash('error', 'Akun Investee sudah didaftarkan');
    return redirect()->back();
}

$validator = Validator::make($request->all(), [
    'name'          => 'required',
    'address'       => 'required',
    'city'          => 'required',
    'province'      => 'required',
    'contact_person' => 'required',
    'contact_no'    => 'required',
]);

if ($validator->fails()) {
    return redirect()->back()->withInput();
}

try{
    Investee::create([
        'student_id'    => $student->id,
        'nama'          => $request->input('name'),
        'address'       => $request->input('address'),
        'city'          => $request->input('city'),
        'province'      => $request->input('province'),
        'fax'           => $request->input('fax'),
        'contact_person' => $request->input('contact_person'),
        'contact_no'    => $request->input('contact_no'),
        'email'         => $student->email,
        'fax'           => $request->input('fax'),
    ]);

    Session::flash('success', 'Akun berhasil didaftarkan');
    return redirect('dashboard/register-status');
}

catch(\Illuminate\Database\QueryException $e)

```



```

    {
        $errorCode = $e->errorInfo[1];
        $errorMsg = $e->errorInfo[2];
        if ($errorCode == 1062) {
            return redirect('/');
        }
        Session::flash('error', $errorMsg);
        return redirect()->back();
    }
}

public function getCreateProjInvestment()
{
    return view('dashboard.pages.investee.create-project-investment');
}

public function postCreateProjInvestment(Request $request){
    $input = $request->all();
    $validator = Validator::make($input, [
        'namainvestasi' => 'required|max:191',
        'description' => 'required',
        'namabank' => 'required|max:191',
        'nomorrekening' => 'required|numeric',
        'atasnama' => 'required|max:191',
        'hargaperlembar' => 'required|numeric',
        'totallembar' => 'required|numeric|min: 1',
        'roi_d' => 'required|numeric',
        'roi_u' => 'required|numeric|gt:roi_d',
        'duedate' => 'required|date',
        'proposalinvestasi' => 'required|mimes:pdf|max:2048',
        'laporankeuangan' => 'required|mimes:pdf|max:2048',
    ]);
    if ($validator->fails()) {
        Session::flash('error', $validator->errors()->first());
    }
}

```

```

        return redirect()->back()->withInput();
    }
    try{
        $berkasinvestasi= $request->file('proposalinvestasi');
        $berkaskeuangan= $request->file('laporankeuangan');
        $studentid= $request->session()->get('id');
        $investeeid = DB::table('investee')
            ->select('investee.id')
            ->where('student_id', '=', $studentid)
            ->Where('status', '=', '1')
            ->first();

        $studentname = $request->session()->get('name');
        $invhash = md5('_INV_'. $studentname. '_'. $request-
>input('description'));

        $keuhash = md5('_KEU_'. $studentname. '_'. $request-
>input('description'));

        $tujuaninv = 'data_files/Student/Investee/Project/proposal_investasi'
;

        $tujuankeu = 'data_files/Student/Investee/Project/lap_keu';
        $extension= 'pdf';
        $filenameinv= "Investment_". $studentid. $investeeid-
>id. $invhash. '._'. $extension;
        $filenamekeu= "Keuangan_". $studentid. $investeeid-
>id. $keuhash. '._'. $extension;
        $berkasinvestasi->move($tujuaninv, $filenameinv);
        $berkaskeuangan->move($tujuankeu, $filenamekeu);
        $formatDate = \Carbon\Carbon::parse($request->input('duedate'))-
>format('Y-m-d');

        Investasi_project::create([
            'nama_investasi'      => $request->input('namainvestasi'),
            'investee_id'         => $investeeid->id,
            'status'              => 0,

```

```

        'status_tempo'           => 0,
        'bank'                   => $request->input('namabank'),
        'no_rekening'            => $request->input('nomorrekening'),
        'atas_nama'              => $request->input('atasnama'),
        'deskripsi_bisnis'       => $request->input('description'),
        'roi_top'                 => $request->input('roi_u'),
        'roi_bot'                 => $request->input('roi_d'),
        'lembar_total'           => $request->input('totallembar'),
        'lembar_terbeli'         => 0,
        'harga_saham'            => $request->input('hargaperlembar'),
        'tgl_jatuh_tempo'        => $formatDate,
        'berkas_proposal_investasi' => $filenameinv,
        'berkas_laporan_keuangan' => $filenamekeu,
    ]);

    Session::flash('success', 'Investasi berhasil didaftarkan, silahkan t
unggu konfirmasi dari pihak admin');

    return view('dashboard.pages.investee.create-project-investment');
}

catch(\Illuminate\Database\QueryException $e)
{
    $errorCode = $e->errorInfo[1];
    $errorMsg = $e->errorInfo[2];

    if ($errorCode == 1062) {
        return redirect('/');
    }

    Session::flash('error', $errorMsg);

    return view('dashboard.pages.investee.create-project-investment');
}
}

public function getCreateFundInvestment()
{
    return view('dashboard.pages.investee.create-funding-investment');
}

```

```

}

public function postCreateFundInvestment(Request $request){

    $input = $request->all();

    $validator = Validator::make($input, [

        'namainvestasi' => 'required|max:191',
        'description'   => 'required',
        'namabank'       => 'required|max:191',
        'nomorrekening' => 'required|numeric',
        'atasnama'       => 'required|max:191',
        'targetdana'    => 'required|numeric',
        'duedate'        => 'required|date',
        'proposalinvestasi' => 'required|mimes:pdf|max:2048',
        'laporankeuangan' => 'required|mimes:pdf|max:2048',

    ]);

    if ($validator->fails()) {

        Session::flash('error', $validator->errors()->first());

        return redirect()->back()->withInput();

    }

    try{

        $berkasinvestasi= $request->file('proposalinvestasi');
        $berkaskeuangan= $request->file('laporankeuangan');
        $studentid= $request->session()->get('id');
        $investeeid = DB::table('investee')
            ->select('investee.id')
            ->where('student_id', '=', $studentid)
            ->Where('status', '=', '1')
            ->first();

        $studentname = $request->session()->get('name');
        $invhash = md5('_INV_'.$studentname.'_').$request-
>input('description'));

        $keuhash = md5('_KEU_'.$studentname.'_').$request-
>input('description'));
    }
}

```

```

        $tujuaninv = 'data_files/Student/Investee/Funding/proposal_investasi'
;

        $tujuankeu = 'data_files/Student/Investee/Funding/lap_keu';
        $extension= 'pdf';

        $filenameinv= "Investment_". $studentid. $investeeid-
>id.$invhash.'.'.$extension;

        $filenamekeu= "Keuangan_". $studentid. $investeeid-
>id.$keuhash.'.'.$extension;

        $berkasinvestasi->move($tujuaninv,$filenameinv);
        $berkaskeuangan->move($tujuankeu,$filenamekeu);

        $formatDate = \Carbon\Carbon::parse($request->input('duedate'))-
>format('Y-m-d');

    Investasi_funding::create([
        'nama_investasi'      => $request->input('namainvestasi'),
        'investee_id'         => $investeeid->id,
        'status'              => 0,
        'status_tempo'        => 0,
        'bank'                 => $request->input('namabank'),
        'no_rekening'         => $request->input('nomorrekening'),
        'atas_nama'           => $request->input('atasnama'),
        'deskripsi_bisnis'    => $request->input('description'),
        'donasi_target'        => $request->input('targetdana'),
        'donasi_masuk'         => 0,
        'tgl_jatuh_tempo'      => $formatDate,
        'berkas_proposal_investasi' => $filenameinv,
        'berkas_laporan_keuangan' => $filenamekeu,
    ]);

    Session::flash('success', 'Investasi berhasil didaftarkan, silahkan t
unggu konfirmasi dari pihak admin');

    return view('dashboard.pages.investee.create-funding-investment');
}

```

```

catch(\Illuminate\Database\QueryException $e)
{
    $errorCode = $e->errorInfo[1];
    $errorMsg = $e->errorInfo[2];
    if ($errorCode == 1062) {
        return redirect('/');
    }
    Session::flash('error', $errorMsg);
    return view('dashboard.pages.investee.create-funding-investment');
}
}

public function submitprogress(Request $request, $id)
{
    $investeeid = $request->session()->get('id');
    $this->validate($request, [
        'berkas_laporan' => 'required|mimes:pdf|max:2048',
    ]);
    $file = $request->file('berkas_laporan');
    $tgl = $request->get('tgl');
    $deskrip = \Carbon\Carbon::now()->format('YmdHs');
    $berkas = 'Progress';
    $file_berkas = $berkas.'_'.$tgl.'_'.$deskrip.'.pdf';
    $tujuan_upload = 'data_files/Student/Investee/Project/Progress';
    $file->move($tujuan_upload,$file_berkas);
    $formatDate = \Carbon\Carbon::parse($request->input('tgl'))->format('Y-m-
d');

    if($request->get('keterangan_tambahan') == NULL)
    {
        $keterangan_tambahan = '-';
    }
    else
    {

```

```

        $keterangan_tambahan = $request->get('keterangan_tambahan');
    }
    try
    {
        $data = array(
            array(
                'investee_id'=> $investeeid,
                'project_id'=> $id,
                'berkas_laporan'=> $file_berkas,
                'tgl' => $formatDate,
                'deskripsi_laporan'=> $request->get('deskripsi_laporan'),
                'keterangan_tambahan'=> $keterangan_tambahan,
                'updated_at' => \Carbon\Carbon::now(),
                'created_at' => \Carbon\Carbon::now(),
            );
            DB::table('progres_project')->insert($data);
            Session::flash('success', 'Berhasil submit progres investasi project!');
;

        return redirect()->back();
    }
    catch(\Illuminate\Database\QueryException $e)
    {
        $errorCode = $e->errorInfo[1];
        $errorMsg = $e->errorInfo[2];
        if ($errorCode == 1062) {
            return redirect('/');
        }
        Session::flash('error', $errorMsg);
        return redirect()->back();
    }
}
}

```

Kode 5.1.5. Kode Dashboard Investee Controller

- **Controllers/IYTController.php**
Controller yang berisi fungsi untuk mengatur mengenai IYT (ITS Youth Technopreneurs)

```
<?php
namespace App\Http\Controllers;
use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;
use Validator;
use App\Investee;
use App\Investasi_project;
use App\Investasi_funding;
use App\Investasi_IYT;
use App\Student;
use App\IYTBatch;
use App\LaporanProgresBulanan;
use App\LaporanKontrolBulanan;
use App\LaporanKemajuan;
use Illuminate\Support\Facades\Redis;
use Carbon\Carbon;
use Session;

class DashboardIYTController extends Controller
{
    public function getHomeIYT(Request $request)
    {
        $id = $request->session()->get('id');
        $iyt = DB::table('investasi_iyt')->where('student_id', '=', $id)-
>first();
        $getyear = explode("-", $iyt->invoice_iyt);
        $year = $getyear[0];
        return view('dashboard.pages.iyt.home')->with('iyt', $iyt)-
>with('year', $year);
    }
}
```



```

    }

    public function getCreateIYT(Request $request)
    {
        $batch = DB::table('i_y_t_batches')
        ->where(function ($query){
            $now= Carbon::now()->format('Y-m-d');
            $query->where('start_date','<=', $now);
            $query->where('end_date','>=', $now);
            $query->where('status','=', '1');
            $query->orderBy('updated_at');
        })
        ->first();

        $id = $request->session()->get('id');
        $IYT = DB::table('investasi_iyt')
        ->leftjoin('i_y_t_batches', 'i_y_t_batches.id', 'investasi_iyt.batch_id')
        ->where('i_y_t_batches.status', '=', '1')
        ->where('investasi_iyt.student_id', '=', $id)
        ->select('*', 'investasi_iyt.id as iyt_id')
        ->first();

        if($IYT){
            return redirect('dashboard/register-IYT-status');
        }
        else{
            if($batch){
                return view('dashboard.pages.student.register-iyt')-
>with('batch', $batch);
            }
            else{
                return view('dashboard.pages.student.register-iyt-not-avail');
            }
        }
    }
}

```

```

public function getRegisterIYTStatus(Request $request)
{
    $id = $request->session()->get('id');
    $IYT = DB::table('investasi_iyt')
        ->leftjoin('i_y_t_batches', 'i_y_t_batches.id', 'investasi_iyt.batch_id')
        ->where('i_y_t_batches.status', '=', '1')
        ->where('investasi_iyt.student_id', '=', $id)
        -
    >select('*', 'investasi_iyt.id as iyt_id', 'investasi_iyt.status as status_iyt')
        ->first();
    if ($IYT){
        if($IYT->status_iyt == 1){
            return redirect('dashboard/IYT');
        }
        else
        {
            $iyt = DB::table('investasi_iyt')->where('id', '=', $IYT-
            >iyt_id)->first();
            return view('dashboard.pages.student.register-iyt-status')-
            >with('iyt', $iyt);
        }
    }
    else{
        return redirect('dashboard/register-IYT');
    }
}

public function postCreateIYT(Request $request){
    $input = $request->all();
    $id = $request->session()->get('id');
    $student = DB::table('students')->find($id);
    $iytCheck = DB::table('investasi_iyt')->where('student_id', '=', $id)-
    >first();

```

```

if ($iytCheck){
    Session::flash('error', 'Akun Investee sudah didaftarkan');
    return redirect()->back();
}

$validator = Validator::make($input, [
    'namaketua' => 'required|max:191',
    'namakelompok' => 'required|max:191',
    'proposalbisnis' => 'required|mimes:pdf|max:2048',
    'pitchdesk' => 'required|mimes:pdf|max:2048',
    'batch' => 'required',
    'tahunmasuk' => 'required|max:4',
    'tahunlulus' => 'required|max:4',
    'semester' => 'required',
    'kategori' => 'required',
]);

if ($validator->fails()) {
    Session::flash('error', $validator->errors()->first());
    return redirect()->back()->withInput();
}

$berkaspropbisnis= $request->file('proposalbisnis');
$berkaspitchdesk= $request->file('pitchdesk');
$studentname = $request->session()->get('name');
$propbisnishash = md5('_PROPOSALBISNIS_'.$studentname.'_'.$request-
>input('namakelompok'));

$pitchdeskhash = md5('_PITCHDESK_'.$studentname.'_'.$request-
>input('namakelompok'));

$tujuaninv = 'data_files/Student/IYT/Proposal Bisnis';
$tujuankeu = 'data_files/Student/IYT/Pitch Desk';
$extension= 'pdf';
// $desc= md5($request->input('description'));

$filenameinv= "ProposalBisnis_".$student-
>id.$propbisnishash.'_'.$extension;

```

```

        $filenamekeu= "PitchDesk_".$student-
>id.$pitchdeskhash.'.'.$extension;

        $berkaspropbisnis->move($tujuaninv,$filenameinv);
        $berkaspitchdesk->move($tujuankeu,$filenamekeu);

        $now = Carbon::now();
        $year = $now->year;
        $no = $this->_nextInvoiceIYTNumber();
        $id_iyt = $year.'-'. $no;
        Investasi_IYT::create([

            'nama_ketua'          => $request->input('namaketua'),
            'student_id'          => $student->id,
            'status'              => 0,
            'nama_kelompok'      => $request->input('namakelompok'),
            'berkas_proposal_bisnis'=> $filenameinv,
            'berkas_pitch_desk'   => $filenamekeu,
            'batch_id'            => $request->input('batch'),
            'invoice_iyt'         => $id_iyt,
            'tahun_masuk'         => $request->input('tahunmasuk'),
            'tahun_keluar'        => $request->input('tahunlulus'),
            'kategori'            => $request->input('kategori'),
            'semester'            => $request->input('semester'),

        ]);

        Session::flash('success', 'Kelompok berhasil didaftarkan dan akan dis
eleksi, silahkan tunggu hasil pengumuman dari pihak admin');

        return redirect('dashboard/register-IYT-status');
    }

    catch(\Illuminate\Database\QueryException $e)
    {

        $errorCode = $e->errorInfo[1];
        $errorMsg = $e->errorInfo[2];

        if ($errorCode == 1062) {

            return redirect('/');
        }
    }

```

```

    }

    Session::flash('error', $errorMsg);

    return redirect()->back();

}

}

```

Kode 5.1.6. Kode IYT Controller

- Controllers/DashboardStudentController.php
Controller yang berisi fungsi-fungsi yang mengatur Dashboard Student

```

<?php
namespace App\Http\Controllers;
use App\Http\Controllers\Controller;
use Illuminate\Support\Facades\DB;
use Illuminate\Http\Request;
use Session;
use App\Student;
use App\Job;
use App\Employer;
use App\Seminar;
use App\Service;
use Validator;
class DashboardStudentController extends Controller
{
    public function getCreateService()
    {
        return view('dashboard.pages.student.create-service');
    }

    public function postCreateService(Request $request)
    {
        // dd($request);
        $this->validate($request, [

```

```

        'name'            => 'required|max:255',
        'category'        => 'required',
        'description'     => 'required|max:255',
    ]);
    try{
        Service::create([
            'name'            => $request->input('name'),
            'description'     => $request->input('description'),
            'job_category_id' => $request->input('category'),
            'student_id'     => $request->session()->get('id'),
            'status'         => 0,
        ]);
        Session::flash('success', 'Jasa berhasil didaftarkan, silahkan tunggu
konfirmasi jasa');
        return view('dashboard.pages.student.create-service');
    }
    catch(\Illuminate\Database\QueryException $e)
    {
        $errorCode = $e->errorInfo[1];
        $errorMsg = $e->errorInfo[2];
        if ($errorCode == 1062) {
            return redirect('/');
        }
        Session::flash('error', $errorMsg);
        return view('dashboard.pages.student.create-service');
    }
}

public function acceptNewApplicants(Request $request, $id)
{
    $acc = DB::table('guest_services')
        ->where('id', $id)
        ->update([

```

```

        'status' => 1,
        'updated_at' => \Carbon\Carbon::now(),
    ]);
    return redirect()->back();
}

public function rejectNewApplicants(Request $request, $id)
{
    $acc = DB::table('guest_services')
        ->where('id', $id)
        ->update([
            'status' => 2,
            'updated_at' => \Carbon\Carbon::now(),
        ]);
    return redirect()->back();
}

public function getServicesApplicantAccepted(Request $request)
{
    $where_done = [
        'student_id' => $request->session()->get('id'),
        'guest_services.status' => '1',
        'guest_services.status_pekerjaan' => '1',
    ];
    $applicants_done = DB::table('guest_services')
        ->join('guests', 'guest_services.guest_id', 'guests.id')
        -
    >join('services', 'guest_services.service_id', 'services.id')
        ->join('students', 'services.student_id', 'students.id')
        -
    >select('guest_services.id as gsid', 'guests.name as guestname', 'services.name as servicename', 'guests.email', 'guests.mobile_no', 'guest_services.status', 'guest_services.status_pekerjaan', 'guest_services.created_at', 'guest_services.updated_at')

```

```

        ->where($where_done)
        ->paginate(20);

$where_notdone = [
    'student_id' => $request->session()->get('id'),
    'guest_services.status' => '1',
    'guest_services.status_pekerjaan' => '0',
];

$applicants_notdone = DB::table('guest_services')
    ->join('guests', 'guest_services.guest_id', 'guests.id')
    -
>join('services', 'guest_services.service_id', 'services.id')
    ->join('students', 'services.student_id', 'students.id')
    -

>select('guest_services.id as gsid', 'guests.name as guestname', 'services.name as servicename', 'guests.email', 'guests.mobile_no', 'guest_services.status', 'guest_services.status_pekerjaan', 'guest_services.created_at', 'guest_services.updated_at')

        ->where($where_notdone)
        ->paginate(20);

// dd($applicants_notdone);

return view('dashboard.pages.student.service-applicant-accepted')-
>with('applicants_done', $applicants_done)-
>with('applicants_notdone', $applicants_notdone);
}

public function getJobsApproval(Request $request)
{
    $where = [
        'job_student.student_id' => $request->session()->get('id'),
    ];

    $jobs = DB::table('job_student')
        ->join('students', 'job_student.student_id', 'students.id')
        ->join('jobs', 'job_student.job_id', 'jobs.id')

```



```

        ->join('employers', 'jobs.employer_id', 'employers.id')
        -
>select('job_student.created_at', 'job_student.updated_at', 'jobs.name as jobname',
        'employers.name as empname', 'employers.email', 'job_student.status')
        ->where($where)
        ->paginate(20);

        return view('dashboard.pages.student.job-approval')->with('jobs', $jobs);
    }

    public function getOrderList(Request $request)
    {
        $where = [
            'id_investor' => $request->session()->get('id'),
            'role' => $request->session()->get('role'),
        ];

        $orders = Order::where($where)->orderBy('updated_at', 'desc')->
>paginate(25);

        return view('dashboard.pages.order-list')->with('orders', $orders);
    }

    public function showProjectListStudent(Request $request)
    {
        $studentid= $request->session()->get('id');
        $investment = DB::table('order')
            ->select('order.*')
            ->where('order.id_investor', '=', $studentid)
            ->where('role', '=', 'student')
            ->where('status', '=', 'paid')
            ->where('tipe_investasi', '=', 'project')
            ->paginate(25);

        return view('dashboard.pages.student.on-going-project-list')->
>with('investment', $investment);
    }

```

Kode 5.1.7. Kode Dashboard Student Controller

- **Controllers/EmployerController.php**
Controller yang berisi fungsi-fungsi untuk mengatur user Employer

```
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Hash;
use Session;
use App\Employer;
use Image;
use Validator;
use Illuminate\Support\Facades\DB;
class EmployerController extends Controller
{
    public function showLogin() {
        ini_set('display_errors', 1);
        ini_set('display_startup_errors', 1);
        error_reporting(E_ALL);
        return view('pages.employer.auth.login-employer');
    }
    // login
    public function Login(Request $request) {
        $employer = Employer::where('email', '=', $request->input('email'))->first();
        if($employer) {
            if($employer->status == 1){
                if(Hash::check($request->input('password'), $employer->password)) {
                    $request->session()->put([
                        'login' => true,
                        'id' => $employer->id,
                        'name' => $employer->name,
```



```

        'province' => 'required',
        'website' => 'nullable',
        'contact_person' => 'required',
        'contact_no' => 'required|max:14',
        'fax' => 'nullable',
        'password' => 'required|min:8|required_with:password_confirmation',
        'password_confirmation' => 'required|min:8|same:password',
    ]);

    if ($validator->fails()) {
        Session::flash('error', $validator->errors()->all());
        return redirect()->back()->withInput();
    }

    try {
        $filelogo = $request->file('logo');
        $email= $request->input('email');
        $extension= $filelogo->getClientOriginalExtension();
        $namafile=md5($email);
        $hashname= $namafile.'.'.$extension;
        $tujuan_upload = 'data_files/Employer/employer_logo';
        $filelogo->move($tujuan_upload,$hashname);
        $berkas= $request->file('berkas_verifikasi');
        $nama= str_replace(' ','_', $request->input('name'));
        $email= md5($request->input('email'));
        $extension= $berkas->getClientOriginalExtension();
        $filename= $nama.'_'.$email.'.'.$extension;
        $tujuan = 'data_files/Employer/berkas_verifikasi';
        $berkas->move($tujuan,$filename);
        Employer::create([
            'name' => $request->input('name'),
            'logo' => $hashname,
            'berkas_verifikasi' => $filename,
            'email' => $request->input('email'),

```

```

        'address' => $request->input('address'),
        'city' => $request->input('city'),
        'province' => $request->input('province'),
        'website' => $request->input('website'),
        'contact_no' => $request->input('contact_no'),
        'contact_person' => $request->input('contact_person'),
        'fax' => $request->input('fax'),
        'status' => 0,
        'password' => Hash::make($request->input('password')),
    ]);

    Session::flash('success', 'Akun berhasil didaftarkan, silahkan menunggu akun untuk diverifikasi');

    return view('pages.employer.login-warning');

}

catch(\Illuminate\Database\QueryException $e)
{
    $errorCode = $e->errorInfo[1];
    $errorMsg = $e->errorInfo[2];
    if ($errorCode == 1062) {
        return redirect('/register-er');
    }

    Session::flash('error', $errorMsg);
    return redirect()->back();
}

}

public function downloadberkas($berkas)
{
    $where = [
        'employers.id' => $berkas,
    ];

    $berkas_db = DB::table('employers')

```

```

-
>select('employers.name as name', 'employers.email as email', 'employers.berkas_v
erifikasi as berkas')
    ->where($where)
    ->first();
    $file = public_path('data_files/Employer/berkas_verifikasi/'.$berkas_db-
>berkas);
    return response()->download($file, $berkas_db->berkas);
}
}

```

Kode 5.1.8. Kode Employee Controller

- **Controllers/GuestController.php**
Controller yang berisi fungsi-fungsi untuk mengatur user Guest

```

<?php
namespace App\Http\Controllers;
use App\Http\Controllers\Controller;
use Illuminate\Support\Facades\Hash;
use Session;
use Illuminate\Http\Request;
use App\Guest;
use Validator;
use Illuminate\Support\Facades\DB;
class GuestController extends Controller
{
    public function showLogin()
    {
        return view('pages.guest.auth.login-guest');
    }
    public function downloadBerkas($berkas)
    {

```

```

$where = [
    'guests.id' => $berkas,
];

$berkas_db = DB::table('guests')
-
>select('guests.name as name', 'guests.email as email', 'guests.berkas_verifikasi
as berkas')
->where($where)
->first();

$file = public_path('data_files/Guest/berkas_verifikasi/'.$berkas_db-
berkas);

return response()->download($file, $berkas_db->berkas);
}

public function Login(Request $request)
{
    $guest = Guest::where('email', '=', $request->input('email'))->first();
    if ($guest) {
        if($guest->status_gs == 1){
            if (Hash::check($request->input('password'), $guest->password)) {
                $request->session()->put([
                    'login' => true,
                    'id' => $guest->id,
                    'name' => $guest->name,
                    'email' => $guest->email,
                    'role' => 'guest',
                ]);

                Session::flash('success', 'Anda berhasil login');
                return redirect('/dashboard');
            }else{
                Session::flash('error', 'Password tidak cocok');
                return redirect()->route('guest.showLogin');
            }
        }
    }
}

```



```

try {
    $berkas= $request->file('berkas');
    $nama= str_replace(' ','_',$request->input('name'));
    $email= md5($request->input('email'));
    $extension= $berkas->getClientOriginalExtension();
    $filename= $nama.'_'.$email.'.'.$extension;
    $tujuan_upload = 'data_files/Guest/berkas_verifikasi';
    $berkas->move($tujuan_upload,$filename);
    Guest::create([
        'email' => $request->input('email'),
        'password' => Hash::make($request->input('password')),
        'name' => $request->input('name'),
        'status' => $request->input('status'),
        'pekerjaan' => $request->input('pekerjaan'),
        'mobile_no' => $request->input('mobile_no'),
        'status_gs' => 0,
        'berkas_verifikasi' => $filename,
    ]);
    Session::flash('success', 'Akun berhasil didaftarkan, silahkan menung
gu akun untuk diverifikasi');
    return view('pages.guest.login-warning');
}
catch(\Illuminate\Database\QueryException $e)
{
    $errorCode = $e->errorInfo[1];
    $errorMsg = $e->errorInfo[2];
    if ($errorCode == 1062) {
        return redirect('/register-gs');
    }
    Session::flash('error', $errorMsg);
    return redirect()->back();
}

```

```
}  
}
```

Kode 5.1.9. Kode Guest Controller

- **Controllers/InvestasiController.php**
Controller yang berisi fungsi-fungsi untuk mengatur fungsional tentang investasi serta penggunaan Midtrans

```
<?php  
namespace App\Http\Controllers;  
use App\Http\Controllers\Controller;  
use Illuminate\Http\Request;  
use Illuminate\Support\Facades\DB;  
use Session;  
use Validator;  
use App\Student;  
use App\Guest;  
use App\Investasi;  
use App\Investasi_project;  
use App\Investasi_funding;  
use App\Order;  
use App\Payment;  
class InvestasiController extends Controller  
{  
  
    public function showProjectIndex()  
    {  
        $investasiCount = DB::table('investasi_project')  
            ->where('status', '=', 1)  
            ->count();  
  
        $investasi = DB::table('investasi_project')
```

```

-
>join('investee', 'investee.id', 'investasi_project.investee_id')
-
>select('investasi_project.*', 'investee.nama as nama_investee')
    ->where('investasi_project.status', '=', 1)
    ->paginate(8);

    return view('investasi-project-list')->with('investasis', $investasi)-
>with('investasiCount', $investasiCount);
}
// midtrans initializer
protected function _initPaymentGateway()
{
    // Set your Merchant Server Key
    \Midtrans\Config::$serverKey = 'SB-Mid-server--5w-DiJiV2WQ_vcgFY1GZZ8Y';
    // Set to Development/Sandbox Environment (default). Set to true for Prod
uction Environment (accept real transaction).
    \Midtrans\Config::$isProduction = false;
    // Set sanitization on (default)
    \Midtrans\Config::$isSanitized = true;
    // Set 3DS transaction for credit card to true
    \Midtrans\Config::$is3ds = true;
}
protected function _nextStudentProjectInvoiceNumber()
{
    $where = [
        'tipe_investasi' => 'project',
        'role' => 'student',
    ];
    $lastOrder = Order::where($where)->orderBy('created_at', 'desc')-
>first();
    if($lastOrder){
        $lastInvoice = explode("/", $lastOrder->invoice);
    }
}

```

```

        return $lastInvoice[1] + 1;
    }else{

        return 1;
    }
}

public function beliSaham(Request $request, $id_inv)
{
    // * cek jumlah lembar yang tersedia
    $investasi = Investasi_project::where('investasi_project.id', '=', $id_in
v)

        ->join('investee', 'investee.id', 'investee_id')
        -
>select('investasi_project.*', 'investee.nama as nama_investee', 'investee.id as
investee_id')

        ->first();

    $lembar_sisa = $investasi->lembar_total - $investasi->lembar_terbeli;
    if ($request->input('lembar_beli') > $lembar_sisa){
        Session::flash('error', 'Tidak dapat membeli sebanyak '.$request-
>input('lembar_beli').', hanya tersisa '.$lembar_sisa.' lembar');
        return redirect()->back();
    }else{
        $investasi->lembar_terbeli = $investasi->lembar_terbeli + $request-
>input('lembar_beli');
        $investasi->save();
    }
    // * id pembeli dari session
    $id_pembeli = $request->session()->get('id');
    // * validasi request
    $validator = Validator::make($request->all(),[
        'lembar_beli' => 'required',

```

```

        'termspolicy' => 'required',
    ));
    if ($validator->fails()){
        Session::flash('error', $validator->errors()->first());
        return redirect()->back();
    }

    // * transaksi role student
    if ($request->session()->get('role') == 'student'){
        // * ambil data student
        $student = Student::where('id', '=', $id_pembeli)->first();
        $this->_initPaymentGateway();

        $redirectSnap = '';

        \DB::transaction(function() use ($request, $student, $investasi, &$re
directSnap){
            // * INV/#/ST/YYYYMMDD/PJ => format
            $nextInvNum = $this->_nextStudentProjectInvoiceNumber();
            $newInvoice = 'INV/'. $nextInvNum . '/ST/' . date("Ymd") . '/PJ';
            $order = Order::create([
                'invoice' => $newInvoice,
                'nama_investor' => $student->name,
                'email_investor' => $student->email,
                'id_investor' => $student->id,
                'role' => 'student',
                'tipe_investasi' => 'project',
                'investasi_id' => $request->input('project_id'),
                'nama_investasi' => $investasi->nama_investasi,
                'nama_investee' => $investasi->nama_investee,
                'id_investee' => $investasi->investee_id,
                'lembar_beli' => $request->input('lembar_beli'),
                'total_harga' => $request->input('total_harga'),
            ]);
        });
    }
}

```

```

        'order_date' => NULL,
        'payment_due' => NULL,
    ]);
    $params = [
        'transaction_details' => [
            'order_id' => $order->invoice,
            'gross_amount' => $order->total_harga,
        ],
        'customer_details' => [
            'first_name' => $order->nama_investor,
            'last_name' => '',
            'email' => $order->email_investor,
            'phone' => $student->mobile_no,
        ],
    ];

    $responseAPI = \Midtrans\Snap::createTransaction($params);
    $redirectSnap = $responseAPI->redirect_url;
    $order->snap_token = $responseAPI->token;
    $order->save();
});

return redirect($redirectSnap);
}else if ($request->session()->get('role') == 'guest'){
    $guest = Guest::where('id', '=', $id_pembeli)->first();
    $this->_initPaymentGateway();
    $redirectSnap = '';
    \DB::transaction(function() use ($request, $guest, $investasi, &$redirectSnap){
        // INV/#/GS/YYYYMMDD/PJ
        $nextInvNum = $this->_nextGuestProjectInvoiceNumber();
        $newInvoice = 'INV/'. $nextInvNum . '/GS/' . date("Ymd") . '/PJ';
        $order = Order::create([

```

```

        'invoice' => $newInvoice,
        'nama_investor' => $guest->name,
        'email_investor' => $guest->email,
        'id_investor' => $guest->id,
        'role' => 'guest',
        'tipe_investasi' => 'project',
        'investasi_id' => $request->input('project_id'),
        'nama_investasi' => $investasi->nama_investasi,
        'nama_investee' => $investasi->nama_investee,
        'id_investee' => $investasi->investee_id,
        'lembar_beli' => $request->input('lembar_beli'),
        'total_harga' => $request->input('total_harga'),
        'order_date' => NULL,
        'payment_due' => NULL,
    ]]);
    $params = [
        'transaction_details' => [
            'order_id' => $order->invoice,
            'gross_amount' => $order->total_harga,
        ],
        'customer_details' => [
            'first_name' => $order->nama_investor,
            'last_name' => '',
            'email' => $order->email_investor,
            'phone' => $guest->mobile_no,
        ],
    ];
    $responseAPI = \Midtrans\Snap::createTransaction($params);
    $redirectSnap = $responseAPI->redirect_url;
    $order->snap_token = $responseAPI->token;
    $order->save();
    });

```

```

        return redirect($redirectSnap);
    }
}

public function donasi(Request $request, $id_inv)
{
    $investasi = Investasi_funding::where('investasi_funding.id', '=', $id_inv)
v)
        ->join('investee', 'investee.id', 'investee_id')
        -
>select('investasi_funding.*', 'investee.nama as nama_investee', 'investee.id as
investee_id')
        ->first();

    $id_pembeli = $request->session()->get('id');
    $validator = Validator::make($request->all(), [
        'total_harga' => 'required',
        'termspolicy' => 'required',
    ]);
    if ($validator->fails()){
        Session::flash('error', $validator->errors()->first());
        return redirect()->back();
    }
    if ($request->session()->get('role') == 'student') {
        // * ambil data student
        $student = Student::where('id', '=', $id_pembeli)->first();
        $this->_initPaymentGateway();
        $redirectSnap = '';
        \DB::transaction(function() use ($request, $student, $investasi, &$re
directSnap){
            // * INV/#/STD/YYYYMMDD/FD => format
            $nextInvNum = $this->_nextStudentFundingInvoiceNumber();

```



```

$newInvoice = 'INV/'. $nextInvNum . '/ST/' . date("Ymd") . '/FD';
$order = Order::create([
    'invoice' => $newInvoice,
    'nama_investor' => $student->name,
    'email_investor' => $student->email,
    'id_investor' => $student->id,
    'role' => 'student',
    'tipe_investasi' => 'funding',
    'investasi_id' => $request->input('project_id'),
    'nama_investasi' => $investasi->nama_investasi,
    'nama_investee' => $investasi->nama_investee,
    'id_investee' => $investasi->investee_id,
    'lembar_beli' => 0,
    'total_harga' => $request->input('total_harga'),
    'order_date' => NULL,
    'payment_due' => NULL,
]);
$params = [
    'transaction_details' => [
        'order_id' => $order->invoice,
        'gross_amount' => $order->total_harga,
    ],
    'customer_details' => [
        'first_name' => $order->nama_investor,
        'last_name' => '',
        'email' => $order->email_investor,
        'phone' => $student->mobile_no,
    ],
];
$responseAPI = \Midtrans\Snap::createTransaction($params);
$redirectSnap = $responseAPI->redirect_url;
$order->snap_token = $responseAPI->token;

```

```

        $order->save();

    });

    return redirect($redirectSnap);
} else if ($request->session()->get('role') == 'guest') {
    $guest = Guest::where('id', '=', $id_pembeli)->first();
    $this->_initPaymentGateway();
    $redirectSnap = '';
    \DB::transaction(function() use ($request, $guest, $investasi, &$redirectSnap) {

        // INV/#/GS/YYYYMMDD/FD

        $nextInvNum = $this->_nextGuestFundingInvoiceNumber();
        $newInvoice = 'INV/' . $nextInvNum . '/GS/' . date("Ymd") . '/FD';
        $order = Order::create([

            'invoice' => $newInvoice,
            'nama_investor' => $guest->name,
            'email_investor' => $guest->email,
            'id_investor' => $guest->id,
            'role' => 'guest',
            'tipe_investasi' => 'funding',
            'investasi_id' => $request->input('project_id'),
            'nama_investasi' => $investasi->nama_investasi,
            'nama_investee' => $investasi->nama_investee,
            'id_investee' => $investasi->investee_id,
            'lembar_beli' => 0,
            'total_harga' => $request->input('total_harga'),
            'order_date' => NULL,
            'payment_due' => NULL,

        ]);

        $params = [

            'transaction_details' => [

                'order_id' => $order->invoice,

```

```

        'gross_amount' => $order->total_harga,
    ],
    'customer_details' => [
        'first_name' => $order->nama_investor,
        'last_name' => '',
        'email' => $order->email_investor,
        'phone' => $guest->mobile_no,
    ],
];
$responseAPI = \Midtrans\Snap::createTransaction($params);
$redirectSnap = $responseAPI->redirect_url;
$order->snap_token = $responseAPI->token;
$order->save();
});
return redirect($redirectSnap);
}

}

protected function _updateHapusLembarBeli($investasi_id, $lembar)
{
    $investasi = Investasi_project::where('id', '=', $investasi_id)->first();
    $investasi->lembar_terbeli = $investasi->lembar_terbeli - $lembar;
    $investasi->save();
}

protected function _updateTambahDonasiMasuk($investasi_id, $donasi)
{
    $investasi = Investasi_funding::where('id', '=', $investasi_id)->first();
    $investasi->donasi_masuk = $investasi->donasi_masuk + $donasi;
    $investasi->save();
}
}

```

Kode 5.1.10. Kode Investasi Controller

- **Controllers/JobController.php**
Controller yang berisi fungsi-fungsi untuk mengatur fungsional tentang Pekerjaan yang ditawarkan employer

```
<?php
namespace App\Http\Controllers;
use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;
use Session;
use App\Job;
use App\JobCategory;
use App\Employer;
class JobController extends Controller
{
    public function index()
    {
        $jobsCount= DB::table('jobs')
            ->where('jobs.status','=',1)
            ->count();

        $jobcategory = DB::table('job_categories')->select('name','slug')->get();
        $job = DB::table('jobs')
            -
        >join('job_categories', 'jobs.job_category_id' ,'job_categories.id')
            -
        >join('employers', 'jobs.employer_id', 'employers.id')
            -
        >select('jobs.expected_salary_high', 'jobs.expected_salary_low', 'jobs.id as id',
        'jobs.name as name' , 'jobs.job_type as job_type','jobs.location as location', 'e
        mployers.name as employername', 'employers.logo as emlogo')
            ->where('jobs.status', '=', 1)
            ->paginate(8);
```

```

        // dd($job);

        return view('job-list')->with('job', $job)->with('jobsCount', $jobsCount)-
>with('jobcategory', $jobcategory);
    }

    public function apply(Request $request, $id)
    {

        $id_stud = $request->session()->get('id');
        $applicant = DB::table('job_student')->where([
            ['job_id', '=', $id],
            ['student_id', '=', $id_stud],
        ]->first();
        if($applicant)
        {
            Session::flash('error', 'Pekerjaan tidak dapat di apply lebih dari satu
kali');
            return redirect()->back();
        }
        else
        {
            $this->validate($request, [
                'cv' => 'required|mimes:pdf|max:2048',
                'motlet' => 'required',
            ]);
            $file = $request->file('cv');
            $cv = 'cv';
            $file_cv = $cv.'-'.$id_stud.'-'.$id.'.pdf';
            $tujuan_upload = 'data_files/Student/Job/CV';
            $file->move($tujuan_upload,$file_cv);
            try
            {

```

```

        $data = array(
            array(
                'student_id'=> $id_stud,
                'job_id'=> $id,
                'status' => 0,
                'cv'=> $file_cv,
                'motivation_letter'=> $request->get('motlet'),
                'updated_at' => \Carbon\Carbon::now(),
                'created_at' => \Carbon\Carbon::now()),
        );

        DB::table('job_student')->insert($data);
        Session::flash('success', 'Berhasil apply job');
        return redirect()->back();
    }
    catch(\Illuminate\Database\QueryException $e)
    {
        $errorCode = $e->errorInfo[1];
        $errorMsg = $e->errorInfo[2];
        if ($errorCode == 1062) {
            return redirect('/');
        }
        Session::flash('error', $errorMsg);
        return redirect()->back();
    }
}
}
}

```

Kode 5.1.11. Kode Job Controller

- Controllers/SeminarController.php
Controller yang berisi fungsi-fungsi untuk fungsional pelaksanaan seminar oleh user Employer

```

namespace App\Http\Controllers;
use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;
use Session;
use App\Seminar;
use App\SeminarCategory;
use App\Employer;
class SeminarController extends Controller
{
    public function index()
    {
        $Count= DB::table('seminars')
            ->where('seminars.status','=',1)
            ->count();

        $seminarcategory = DB::table('seminar_categories')->select('name','slug')->get();

        $seminar = DB::table('seminars')
            -
            >join('seminar_categories', 'seminars.seminar_category_id' , 'seminar_categories.id')
            -
            >join('employers', 'seminars.employer_id', 'employers.id')
            -
            >select('seminars.fee' , 'seminars.id as id','seminars.name as name','seminars.location as location', 'employers.name as employername', 'employers.logo as employerlogo')
            ->where('seminars.status', '=', 1)
            ->paginate(8);
    }
}

```

```

        return view('seminar-list')->with('seminar', $seminar)-
>with('Count', $Count)->with('seminarcategory', $seminarcategory);
    }
}

```

Kode 5.1.12. Kode Seminar Controller

- **Controllers/ServicesController.php**
Controller yang berisi fungsi-fungsi untuk fungsional penawaran Jasa oleh user Guest

```

<?php
namespace App\Http\Controllers;
use App\Http\Controllers\Controller;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;
use Illuminate\Support\Facades\Session;
use App\Services;
use App\JobCategory;
use App\Students;
class ServicesController extends Controller
{
    public function index()
    {
        $servicesCount= DB::table('services')
            ->where('services.status','=',1)
            ->count();

        $jobcategory = DB::table('job_categories')
            ->select('name','slug')->get();

        $services = DB::table('services')
            ->join('students','services.student_id','students.id')
            -
            >join('job_categories','services.job_category_id','job_categories.id')
    }
}

```



```

-
>select('services.id as id','services.name as name', 'students.name as penyedia',
'job_categories.name as category','services.updated_at as lastupdate')
    ->where('services.status','=',1)
    ->paginate(8);

    return view('services-
list',compact('servicesCount','jobcategory','services'));
}

public function approach(Request $request, $id)
{
    $id_guest = $request->session()->get('id');
    $applicant = DB::table('guest_services')->where([
        ['service_id', '=', $id],
        ['guest_id', '=', $id_guest]
    ])->first();
    if($applicant)
    {
        Session::flash('error', 'Sudah anda approach!');
        return redirect()->back();
    }
    else
    {
        try
        {
            $data = array(
                array(
                    'guest_id'=> $id_guest,
                    'service_id'=> $id,
                    'status' => 0,
                    'status_pekerjaan'=>0,
                    'updated_at' => \Carbon\Carbon::now(),
                    'created_at' => \Carbon\Carbon::now(),

```

```

        ),
    );
    DB::table('guest_services')->insert($data);
    Session::flash('success', 'Berhasil apply jasa');
    return redirect()->back();
}
catch(\Illuminate\Database\QueryException $e)
{
    $errorCode = $e->errorInfo[1];
    if ($errorCode == 1062) {
        return redirect('/');
    }
    Session::flash('error', $errorCode);
    return redirect()->back();
}
}
}
}
}

```

Kode 5.1.13. Kode Services Controller

- Controllers/StudentController.php
Controller yang berisi fungsi-fungsi untuk mengatur user Student

```

<?php
namespace App\Http\Controllers;
use App\Http\Controllers\Controller;
use Illuminate\Support\Facades\Hash;
use Illuminate\Http\Request;
use Session;
use Validator;
use App\Student;
use Illuminate\Support\Facades\DB;

```

```

use Carbon\Carbon;

class StudentController extends Controller
{
    public function showLogin()
    {
        return view('pages.student.auth.login-student');
    }

    public function showRegister()
    {
        return view('pages.student.auth.register-student');
    }

    public function Login(Request $request)
    {
        $student = Student::where('email' , '=', $request->input('email'))->first();

        if ($student) {
            if($student->status == 1){
                if (Hash::check($request->input('password'), $student->password)) {
                    $isInvestee = DB::table('investee')->where('student_id', '=', $student->id)
                    ->Where('status', '=', '1')
                    ->first();

                    $IYT = DB::table('investasi_iyt')
                    -
                    >leftjoin('i_y_t_batches', 'i_y_t_batches.id', 'investasi_iyt.batch_id')
                    ->where('i_y_t_batches.status', '=', '1')
                    -
                    >where('investasi_iyt.student_id', '=', $student->id)

```

```
-  
>select('*', 'investasi_iyt.status as status_iyt', 'investasi_iyt.kategori as tin  
gkat_iyt')
```

```
        ->first();  
$now = Carbon::now();  
$dn = $now->toDateString();  
if ($isInvestee)  
    $isInvestee = true;  
else  
    $isInvestee = false;  
if ($IYT){  
    $invoice= $IYT->invoice_iyt;  
    $tingkat_iyt = $IYT->tingkat_iyt;  
    if($IYT->status_iyt == 1){  
        $isIYT = true;  
    }else{  
        $isIYT = false;  
    }  
}  
else{  
    $invoice = 0;  
    $tingkat_iyt = 0;  
    $isIYT = false;  
}  
// dd($isActive);  
$request->session()->put([  
    'login' => true,  
    'id' => $student->id,  
    'name' => $student->name,  
    'email' => $student->email,  
    'role' => 'student',  
    'investee' => $isInvestee,
```

```

        'invoice' => $invoice,
        'iyt' => $isIVT,
        'tingkat_iyt' => $tingkat_iyt,
        // 'batch' => $isActive,
    ]);
    Session::flash('success', 'Anda berhasil Login');
    return redirect('/dashboard');
}
else{
    Session::flash('error', 'Password tidak cocok');
    return redirect()->route('student.showLogin');
}
}
else if($student->status == 0){
    return view('pages.employer.login-warning');
}
else if($student->status == 2){
    return view('pages.employer.login-reject');
}
}
}
else{
    Session::flash('error', 'Akun tidak ditemukan');
    return redirect()->route('student.showLogin');
}
}
return view('pages.student.auth.register-student');
}

public function Register(Request $request)
{
    $formatDate = \Carbon\Carbon::parse($request->input('birthday'))-
>format('Y-m-d');

    $validator = Validator::make($request->all(), [
        'email' => 'required|email|unique:students',
        'password' => 'required|min:8',
        'password_confirmation'=>'required|min:8|same:password',
        'name' => 'required',
    ]);

```

```

        'nrp'            => 'required|min:14|max:14|unique:students',
        'gender'         => 'required',
        'birthday'       => 'required',
        'mobile_no'      => 'required|min:10|max:14',
        'address'        => 'required|max:255',
        'city'           => 'required|max:255',
        'province'       => 'required|max:255',
        'hobby'          => 'required|max:255',
        'skill'          => 'required|max:255',
        'achievement'    => 'required|max:255',
        'experience'     => 'required|max:255',
        'berkas_verifikasi' => 'required|mimes:pdf|max:2048',
    ]);

    if ($validator->fails()) {
        Session::flash('error', $validator->errors()->all());
        return redirect()->back()->withInput();
    }

    try {
        $berkas = $request->file('berkas_verifikasi');
        $nama = str_replace(' ', '_', $request->input('name'));
        $email = md5($request->input('email'));
        $ext = $berkas->getClientOriginalExtension();
        $filename = $nama.'_'.$email.'.'.$ext;
        $tujuan = 'data_files/Student/berkas_validasi';
        Student::create([
            'email'            => $request->input('email'),
            'password'         => Hash::make($request->input('password')),
            'name'             => $request->input('name'),
            'nrp'             => $request->input('nrp'),
            'gender'          => $request->input('gender'),
            'birthdate'       => $formatDate,
            'mobile_no'       => $request->input('mobile_no'),

```

```

        'address'      => $request->input('address'),
        'city'         => $request->input('city'),
        'province'     => $request->input('province'),
        'hobby'        => $request->input('hobby'),
        'skill'        => $request->input('skill'),
        'achievement'  => $request->input('achievement'),
        'experience'   => $request->input('experience'),
        'berkas_validasi' => $filename,
    ]]);
    $berkas->move($tujuan, $filename);
    Session::flash('success', 'Akun berhasil didaftarkan');
    return view('pages.student.auth.login-student');
}

catch(\Illuminate\Database\QueryException $e)
{
    $errorCode = $e->errorInfo[1];
    $errorMsg = $e->errorInfo[2];
    if ($errorCode == 1062) {
        return redirect('/register-st');
    }
    Session::flash('error', $errorMsg);
    return view('pages.student.auth.register-student');
}
}
}

```

Kode 5.1.14. Kode Student Controller

5.2 Implementasi Antarmuka Pengguna

Berikut ini adalah implementasi antarmuka pengguna aplikasi ITS JobX

Register Akun Employer

Nama*
Nama dosen / perusahaan

Email*
Alamat email

Password*
Password

Konfirmasi Password*
Password

Alamat*
Alamat

Gambar 5.2.1 Halaman Pendaftaran User Employer

ITS JobX

Home Portal utama

Create new job

Deskripsi Umum Job

Nama pekerjaan
Nama Pekerjaan

Kategori
Pertanian

Tipe pekerjaan
Freelance

Posisi Pekerjaan/jabatan
Posisi Pekerjaan/jabatan

Lokasi Pekerjaan
Lokasi Pekerjaan

Deskripsi Pekerjaan
Deskripsi Pekerjaan

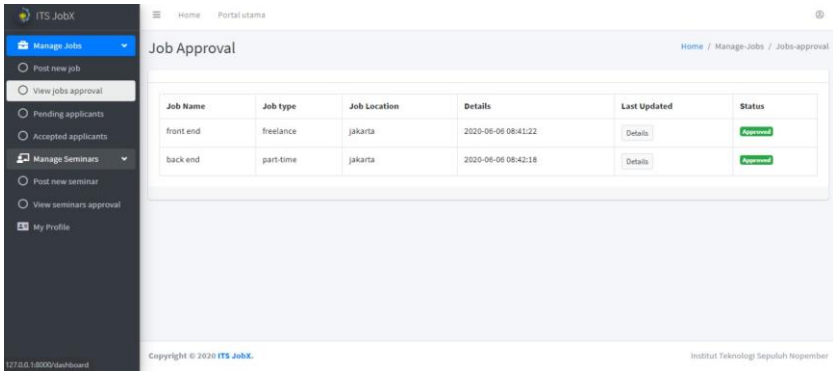
Deskripsi Skill

Minimal Qualification
Kualifikasi minimal pekerjaan

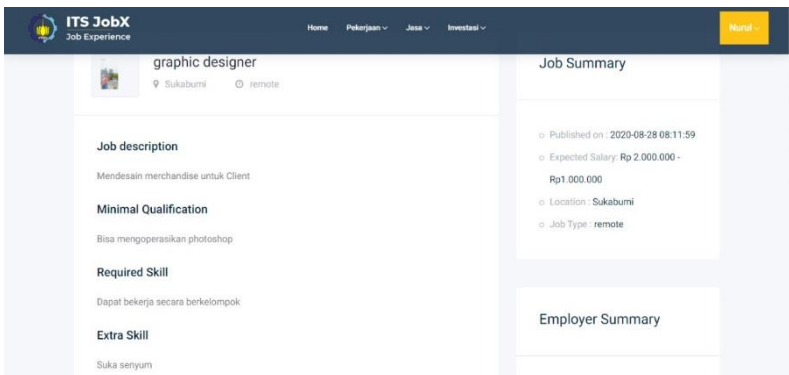
Required skill
Kemampuan yang dibutuhkan untuk pekerjaan

Extra Skill
Kemampuan tambahan yang dapat membantu pekerjaan

Gambar 5.2.2 Halaman Pembuatan Tawaran Pekerjaan



Gambar 5.2.3 Halaman Verifikasi Pekerjaan



Gambar 5.2.4 Halaman Pekerjaan yang ditawarkan

ITS JobX
Job Experience

Home Pekerjaan Jasa Investasi

Nurul

Apply for the job

Upload CV

No file chosen

Motivation Letter

Apply Now

Gambar 5.2.5 Halaman Melamar Pekerjaan bagi Mahasiswa

ITS JobX Admin

Home

Manage Students

Manage Services

Manage Employers

Manage Guests

Manage Investeees

Manage Jobs

Manage Seminars

Manage Project Inv.

Manage Funding Inv.

Approved Jobs

Home / Manage-Jobs / Approved-Jobs

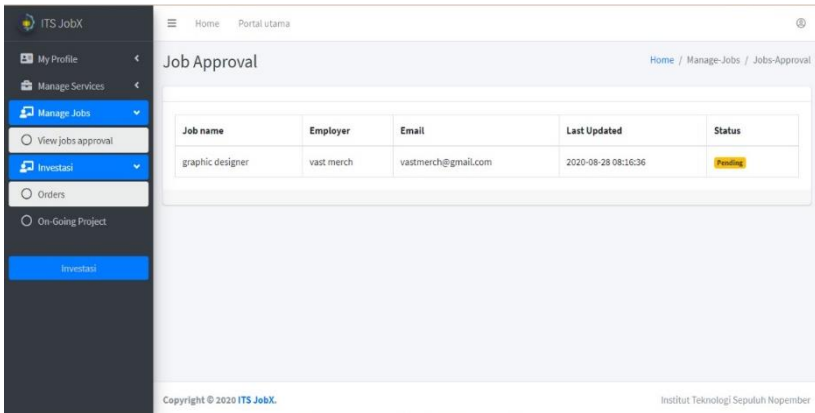
Approved Job Table

Job Name	Job Employer	Job type	Job Location	Details	Action
graphic designer	vast merch	remote	Sukabumi	Details	Reject

Copyright © 2020 ITS JobX

Industri Teknolologi Semesta Nusantara

Gambar 5.2.6 Halaman Cek Pekerjaan yang disetujui ditampilkan



Gambar 5.2.7 Halaman Cek Lamaran Pekerja

ITS JobX

Home Portal utama

Create new seminar

Home / Manage-Seminars / Create-Seminar

Deskripsi Umum Seminar

Nama Seminar

Lokasi

Kategori

Deskripsi Seminar

Surat Bukti penyewaan Tempat*

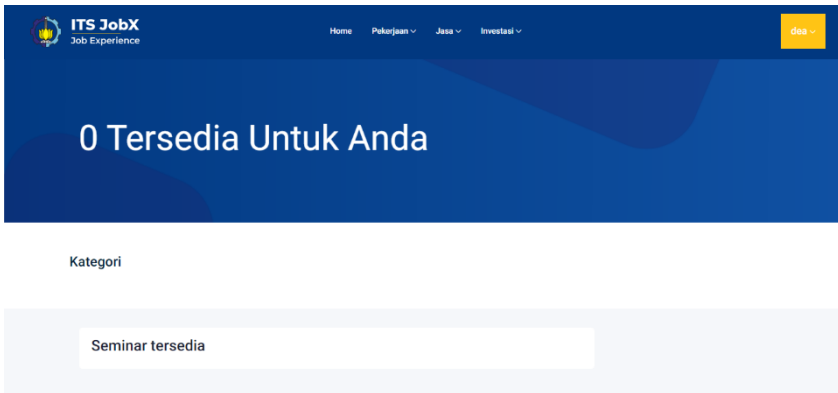
Kontak dan biaya

Contact Person

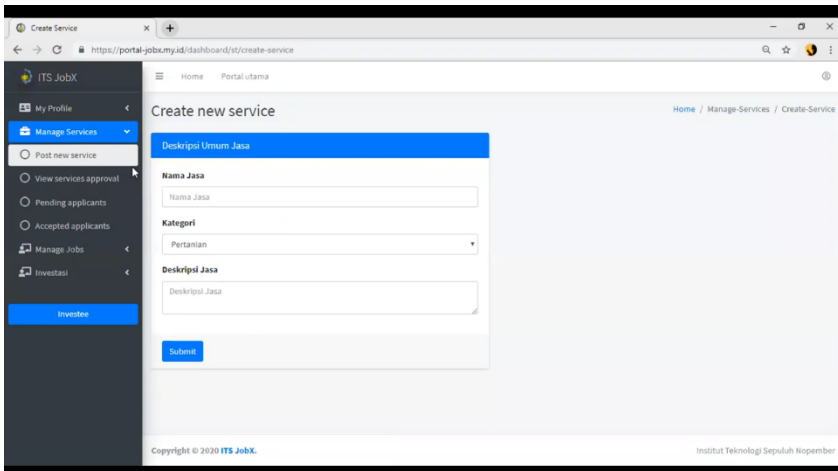
Contact Number

Fee

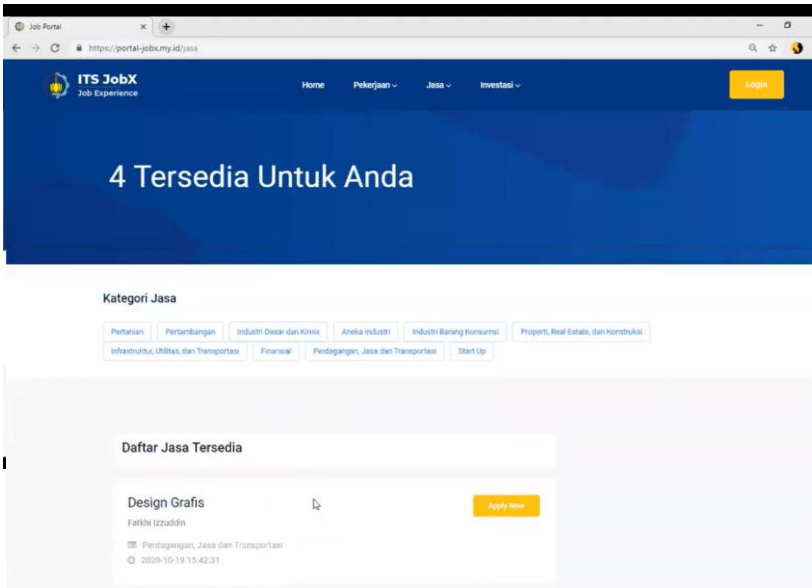
Gambar 5.2.8 Halaman Pengajuan Pelaksanaan Seminar



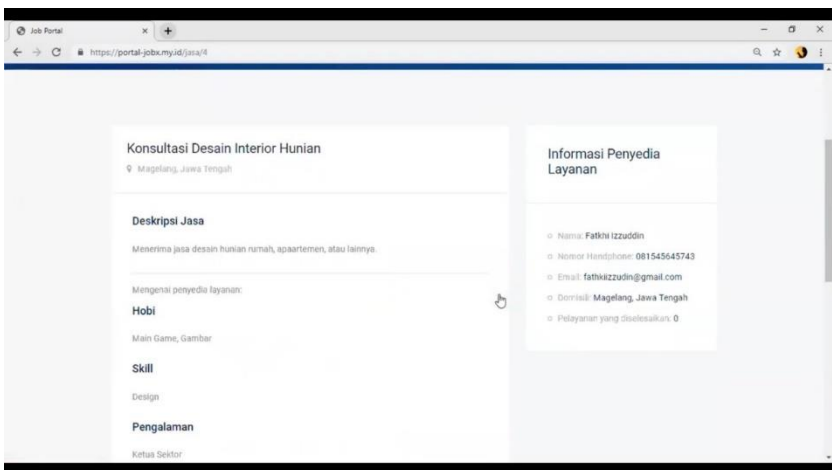
Gambar 5.2.9 Halaman Cek Seminar yang tersedia



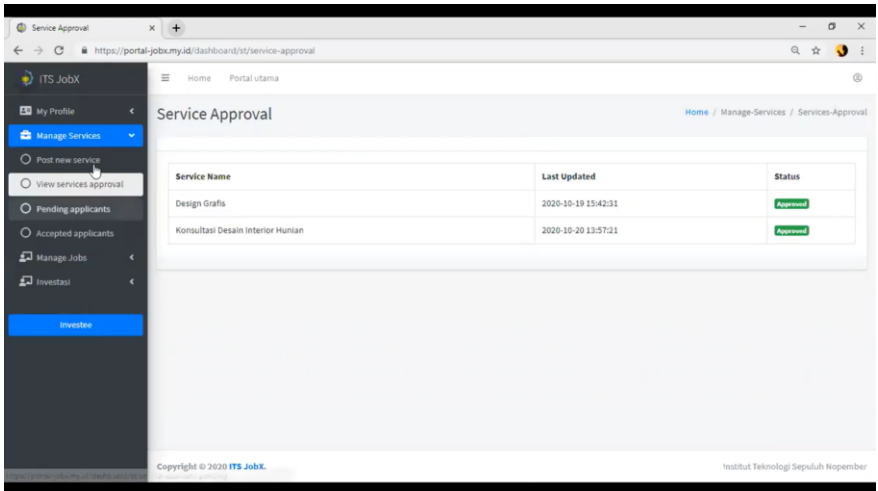
Gambar 5.2.10 Halaman Pembuatan Jasa



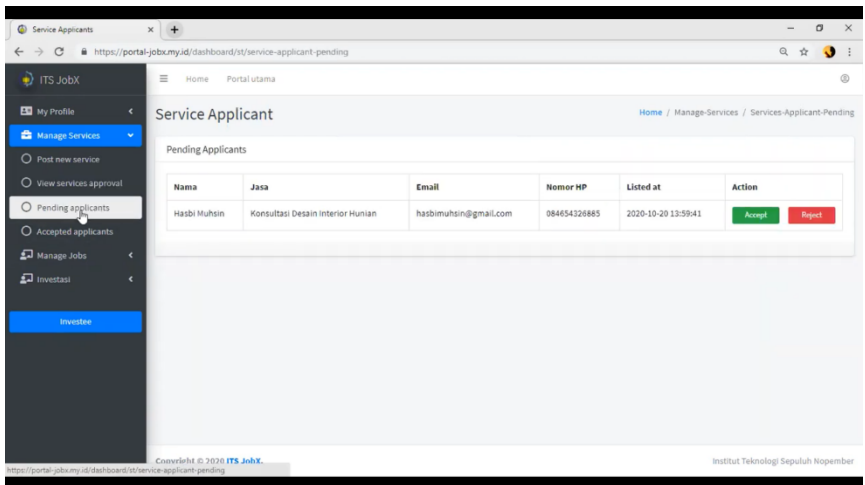
Gambar 5.2.11 Halaman Jasa Tersedia



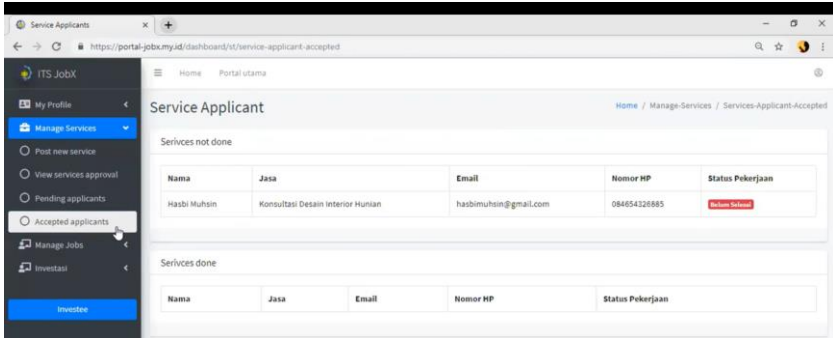
Gambar 5.2.12 Halaman Detail Jasa



Gambar 5.2.13 Halaman Persetujuan Jasa



Gambar 5.2.14 Halaman Permintaan Jasa Masuk



Gambar 5.2.15 Halaman Permintaan Jasa yang sedang berjalan



Gambar 5.2.16 Halaman ITS Youth Technopreneur

[Halaman ini sengaja dikosongkan]

BAB VI

PENGUJIAN DAN EVALUASI

Bab ini menjelaskan tahap uji coba dan evaluasi yang dilakukan terhadap aplikasi website ITS JobX.

6.1 Tujuan Pengujian

Pengujian dilakukan terhadap aplikasi website ITS JobX untuk menguji guna menguji kesesuaian dan ketepatan fungsionalitas dari seluruh sistem aplikasi. serta dapat memenuhi tujuan utama dari aplikasi ini yaitu

6.2 Kriteria Pengujian

Penilaian atas pencapaian tujuan pengujian didapatkan dengan memerhatikan kemampuan aplikasi untuk bisa melakukan fungsi dari masing masing jenis user yaitu user admin, mahasiswa, guest, employer, dan mentor.

6.3 Skenario Pengujian

Pengujian dilakukan dengan mengujikan fungsi utama yang ada pada aplikasi untuk masing masing user yaitu sebagai berikut:

a. Admin

- Melakukan login ke sistem
- Melihat daftar, menyetujui dan menolak user mahasiswa, employer dan guest yang melakukan registrasi ke sistem
- Melihat daftar, menyetujui dan menolak pekerjaan serta seminar yang diajukan oleh user employer ke sistem
- Melihat daftar, menyetujui dan menolak jasa yang diajukan oleh user mahasiswa ke sistem
- Melihat daftar, menyetujui dan menolak investasi yang diajukan oleh mahasiswa ke sistem
- Melihat daftar, menyetujui dan menolak pendaftaran ITS Youth Technopreneur yang diajukan oleh mahasiswa ke sistem

b. Mahasiswa

- Melakukan pendaftaran akun dan login ke sisten
- Melihat serta melakukan apply ke pekerjaan yang didaftarkan ke sistem oleh employer
- Melihat serta mendaftar seminar yang sudah didaftarkan ke sistem oleh employer
- Mendaftarkan jasa yang dimiliki ke sistem
- Melakukan persetujuan ataupun penolakan pemesanan terhadap jasa yang sudah didaftarkan

- Melakukan pendaftaran ITS Youth Technopreneur
 - Melakukan pendaftaran investasi terhadap usaha yang dimiliki
- c. Employer
- Melakukan pendaftaran akun dan login ke sistem
 - Mendaftarkan pekerjaan yang akan ditawarkan ke mahasiswa
 - Mendaftarkan seminar yang akan ditawarkan ke mahasiswa
 - Melakukan seleksi dan persetujuan terhadap apply pekerjaan yang masuk dari mahasiswa
- d. Guest
- Melakukan pendaftaran akun dan login ke sistem
 - Melakukan pemesanan terhadap jasa yang ditawarkan oleh mahasiswa
 - Melakukan investasi terhadap usaha yang dimiliki oleh mahasiswa
- e. Mentor
- Melakukan pendaftaran akun dan login ke sistem
 - Melakukan penjadwalan mentoring dengan peserta IYT
 - Melihat dokumentasi serta notulensi mentoring

6.4 Evaluasi Pengujian

Hasil pengujian dilakukan terhadap pengamatan mengenai perilaku sistem aplikasi website ITS JobX terhadap kasus skenario uji coba. Pengujian dilakukan oleh pihak pengembang, pengguna, dan pembimbing lapangan. Tabel 6.4.1 menjelaskan hasil uji coba terhadap aplikasi yang telah dibuat

Role User	Skenario Pengujian	Hasil Pengujian (Status)
Admin	Melakukan login ke sistem	Terpenuhi
	Melihat daftar, menyetujui dan menolak user mahasiswa, employer dan guest yang melakukan registrasi ke sistem	Terpenuhi
	Melihat daftar, menyetujui dan menolak pekerjaan serta seminar yang diajukan oleh user employer ke sistem	Terpenuhi
	Melihat daftar, menyetujui dan menolak jasa yang diajukan oleh user mahasiswa ke sistem	Terpenuhi

	Melihat daftar, menyetujui dan menolak investasi yang diajukan oleh mahasiswa ke sistem	Terpenuhi
	Melihat daftar, menyetujui dan menolak pendaftaran ITS Youth Technopreneur yang diajukan oleh mahasiswa ke sistem	Terpenuhi
Mahasiswa	Melakukan pendaftaran akun dan login ke sisten	Terpenuhi
	Melihat serta melakukan apply ke pekerjaan yang didaftarkan ke sistem oleh employer	Terpenuhi
	Melihat serta mendaftar seminar yang sudah didaftarkan ke sistem oleh employer	Terpenuhi
	Mendaftarkan jasa yang dimiliki ke sistem	Terpenuhi
	Melakukan persetujuan ataupun penolakan pemesanan terhadap jasa yang sudah didaftarkan	Terpenuhi

	Melakukan pendaftaran ITS Youth Technopreneur	Terpenuhi
	Melakukan pendaftaran investasi terhadap usaha yang dimiliki	Terpenuhi
Employer	Melakukan pendaftaran akun dan login ke sistem	Terpenuhi
	Mendaftarkan pekerjaan yang akan ditawarkan ke mahasiswa	Terpenuhi
	Mendaftarkan seminar yang akan ditawarkan ke mahasiswa	Terpenuhi
	Melakukan seleksi dan persetujuan terhadap apply pekerjaan yang masuk dari mahasiswa	Terpenuhi
Guest	Melakukan pendaftaran akun dan login ke sistem	Terpenuhi
	Melakukan pemesanan terhadap jasa yang ditawarkan oleh mahasiswa	Terpenuhi

	Melakukan investasi terhadap usaha yang dimiliki oleh mahasiswa	Terpenuhi
Mentor	Melakukan pendaftaran akun dan login ke sistem	Terpenuhi
	Melakukan penjadwalan mentoring dengan peserta IYT	Terpenuhi
	Melihat dokumentasi serta notulensi mentoring	Terpenuhi

Tabel 6.4.1 Evaluasi Pengujian

BAB VII

KESIMPULAN DAN SARAN

7.1 Kesimpulan

Kesimpulan yang didapat setelah melaksanakan pembuatan aplikasi ITS JobX pada kegiatan kerja praktik di Pusat Pengembangan Karir dan Kewirausahaan Mahasiswa Institut Teknologi Sepuluh Nopember adalah sebagai berikut:

- Adanya Aplikasi ITS JobX dapat memudahkan mahasiswa untuk mencari pengalaman dengan magang di perusahaan yang sudah terdaftar di ITS JobX, selain itu mahasiswa dapat menawarkan jasa yang dimilikinya untuk bisa disewa atau dibayar oleh pengguna umum lain.
- Framework yang digunakan pada aplikasi ini merupakan framework yang umum, sehingga mudah untuk dapat diperbarui dan digunakan kembali

7.2 Saran

Dalam pengerjaannya, sistem yang dibuat masih memiliki kekurangan, saran untuk pengembangan Aplikasi ITS JobX selanjutnya adalah sebagai berikut:

- Mengintegrasikan data mahasiswa ITS dengan sistem, sehingga, mahasiswa tidak perlu mendaftarkan akunnya, cukup login dengan data seperti NRP dan Password Integra
- Membuat tampilan halaman yang lebih konsisten dimana pada beberapa halaman di Aplikasi ITS JobX yang sekarang masih ada yang menggunakan bahasa Indonesia, dan ada juga yang menggunakan bahasa Inggris.

DAFTAR PUSTAKA

- [1] Beon Intermedia, “Apa itu Laravel? Fitur dan Kelebihan Laravel dibanding Framework lainnya,” 5 February, 2020. [Online]. Available: <https://www.jagoanhosting.com/blog/framework-laravel/>. [Accessed 13 June 2021]
- [2] Gusmi Tasari, “Apa itu Laravel? Mengenal Visual Studio Code,” 25 April 2021. [Online]. Available: <https://www.gamelab.id/news/468-mengena-visual-studio-code>. [Accessed 13 June 2021]
- [3] Yasin K, “Pengertian MySQL, Fungsi, dan Cara Kerjanya (Lengkap),” 24 July 2019. [Online]. Available: <https://www.niagahoster.co.id/blog/mysql-adalah/>. [Accessed 16 June 2021]
- [4] Faradilla A, “Apa Itu PHP? Pengertian dan Tutorial PHP untuk Pemula,” 1 July 2021. [Online]. Available: <https://www.hostinger.co.id/tutorial/apa-itu-php/>. [Accessed 28 June 2021]
- [5] Benefita, “Belajar HTML Lengkap untuk Pemula,” 21 October 2020. [Online]. Available: <https://www.niagahoster.co.id/blog/belajar-html/>. [Accessed 28 June 2021]
- [6] Toni Haryanto, “Pengenalan dan Sintaks Dasar CSS,” 13 July 2016. [Online]. Available: <https://www.codepolitan.com/pengenalan-dan-sintaks-dasar-css>. [Accessed 28 June 2021]
- [7] Rian Romadhon, “Pengertian JavaScript Yang Mudah Untuk Dipahami,” 7 June 2016. [Online]. Available: <https://www.softwareseni.co.id/blog/pengertian-javascript-yang-mudah-untuk-dipahami>. [Accessed 28 June 2021]
- [8] Humaira Aliya, “Kenalan dengan Git, Tools yang Lancarkan Workflow Programmer dan Developer,” 12 February, 2021.

- [Online]. Available: <https://glints.com/id/lowongan/apa-itu-git-adalah/#.YOHf8egzbcc>. [Accessed 31 June 2021]
- [9] Salma Awwabin, "Payment Gateway: Metode Pembayaran untuk Bisnis Online," 23 November, 2020. [Online]. Available: https://www.niagahoster.co.id/blog/payment-gateway/#2_Midtrans. [Accessed 31 June 2021]

BIODATA PENULIS

Nama : Iqbaal Pratama Putra
Tempat, Tanggal Lahir : Jember, 20 Mei 2000
Jenis Kelamin : Laki-Laki
Agama : Islam
Status : Belum Menikah
Alamat : Jl. Letjen Suprpto I/41 Jember
Telepon : 0895399873795
Email : iqbaalpratama23@gmail.com

PENDIDIKAN FORMAL

2018 – sekarang : S1 Teknik Informatika ITS
2015 – 2018 : SMAN 1 Jember
2012 – 2015 : SMPN 2 Jember
2006 – 2012 : SDN Kepatihan 05 Jember

KEMAMPUAN

- *Web Programming* (HTML, PHP, CSS, Javascript)
- *Programming* (C, C++, Python)
- *Database Management* (MySQL)
- *Software* Perkantoran (Microsoft Word, Excel, PowerPoint)
- Bahasa (Indonesia, Inggris)

AKADEMIS

Kuliah : Departemen Teknik Informatika – Fakultas
Teknologi Elektro dan Informatika Cerdas, Institut
Teknologi Sepuluh Nopember Surabaya
Angkatan : 2018
Semester : 6 (Enam)
IPK : 3.72

[Halaman ini sengaja dikosongkan]